The Existing DCT-Based JPEG Standard

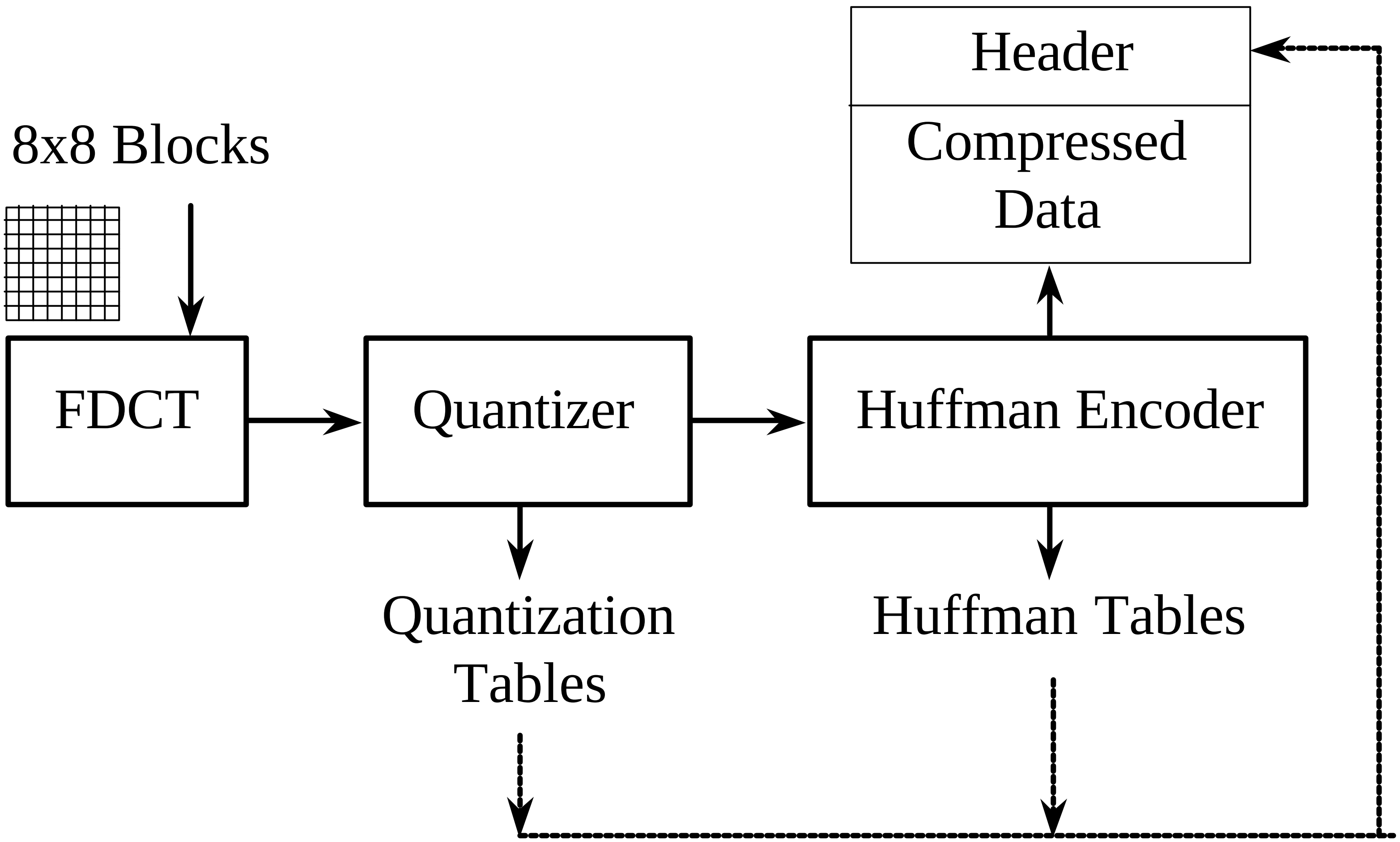
What Is JPEG?

- The JPEG (**Joint Photographic Experts Group**) committee, formed in 1986, has been chartered with the
  - “*Digital compression and coding of continuous-tone still images*”
- Joint between ISO and ITU-T
- Has developed standards for the lossy, lossless, and nearly lossless of still images in the past decade
- Website: **www.jpeg.org**
- The JPEG committee has published the following standards:
  - ISO/IEC 10918-1 | ITU-T Rec. T.81 : *Requirements and guidelines*
  - ISO/IEC 10918-2 | ITU-T Rec. T.83 : *Compliance testing*
  - ISO/IEC 10918-3 | ITU-T Rec. T.84: *Extensions*
  - ISO/IEC 10918-4 | ITU-T Rec. T.86: *Registration of JPEG Parameters, Profiles, Tags, Color Spaces, APPn Markers Compression Types, and Registration Authorities (REGAUT)*

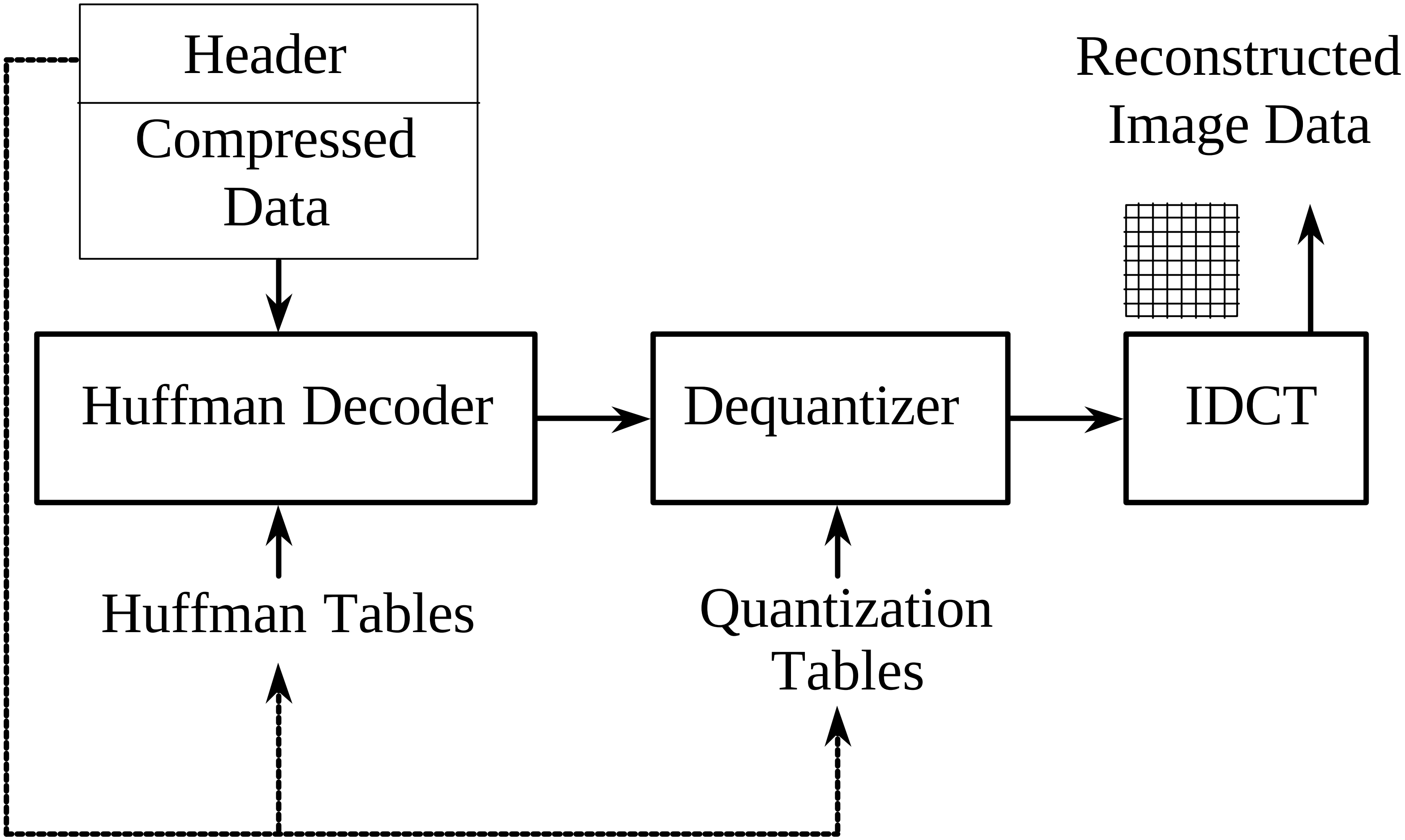
The Existing JPEG Standard “Toolkit”

- The existing **JPEG** standard concerns with the compression of continuous-tone, still-frame, monochrome and color images. It provides a "toolkit" of compression techniques from which applications can select the elements that satisfy their particular requirements.
  - **Baseline system**: A simple and efficient DCT-based algorithm that uses Huffman coding, operates only in sequential mode, and is restricted to 8 bits/pixel input.
  - **Extended system**: Enhancements to the baseline to satisfy broader applications.
  - **Lossless mode**: Based on predictive coding and independent of the DCT that uses either Huffman or arithmetic coding.

JPEG Encoder Block Diagram



JPEG Decoder Block Diagram



Header

Compressed Data

Huffman Decoder

Dequantizer

IDCT

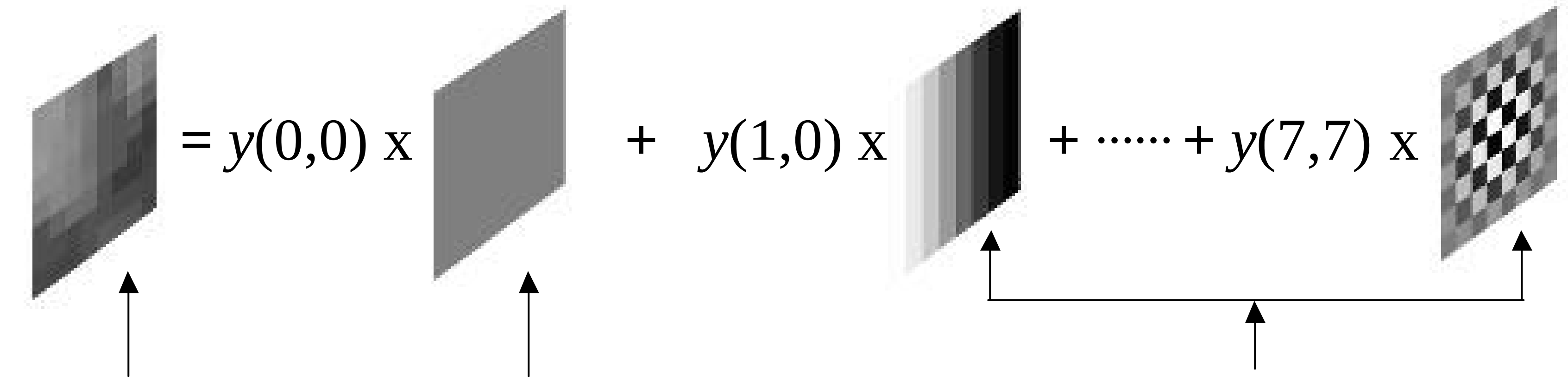
Reconstructed Image Data

Huffman Tables

Quantization Tables

Image Representation with DCT

- DCT coefficients can be viewed as weighting functions that, when applied to the 64 cosine basis functions of various spatial frequencies (8 x 8 templates), will reconstruct the original block.



$= y(0,0) \times \quad + \quad y(1,0) \times \quad + \cdots\cdots + y(7,7) \times$

Original image block

DC (flat) basis function

AC basis functions

JPEG DCT Example

Original Image

139	144	149	153	155	155	155	155
144	151	153	156	159	156	156	156
150	155	160	163	158	156	156	156
159	161	162	160	160	159	159	159
159	160	161	162	162	155	155	155
161	161	161	161	160	157	157	157
162	162	161	163	162	157	157	157
162	162	161	161	163	158	158	158

8 BPP
64 pixels
512 bits

DCT Transformed Image

235.6	-1.0	-12.1	-5.2	2.1	-1.7	-2.7	1.3
-22.6	-17.5	-6.2	-3.2	-2.9	-0.1	0.4	-1.2
-10.9	-9.3	-1.6	1.5	0.2	-0.9	-0.6	-0.1
-7.1	-1.9	0.2	1.5	0.9	-0.1	0.0	0.3
-0.6	-0.8	1.5	1.6	-0.1	-0.7	0.6	1.3
1.8	-0.2	1.6	-0.3	-0.8	1.5	1.0	-1.0
-1.3	-0.4	-0.3	-1.5	-0.5	1.7	1.1	-0.8
-2.6	1.6	-3.8	-1.8	1.9	1.2	-0.6	-0.4

Quantized/ Scaled Transformed Data

15	0	-1	0	0	0	0	0
-2	-1	0	0	0	0	0	0
-1	-1	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

DCT Coefficient Quantization

- Each DCT coefficient is uniformly quantized with a quantization step that is taken from a user-defined **quantization table** (q-table or normalization matrix), characterized by 64, 1-byte elements.

- The quality and compression ratio of an encoded image can be varied by changing the q-table elements (usually by scaling up or down the values of an initial q-table).

- The q-table is often designed according to the perceptual importance of the DCT coefficients (e.g., by using the HVS CSF data) under the intended viewing conditions.

- For the baseline system, in order to meet the needs of the various color components, up to four different quantization tables are allowed.

Example of Luminance Quantization Table

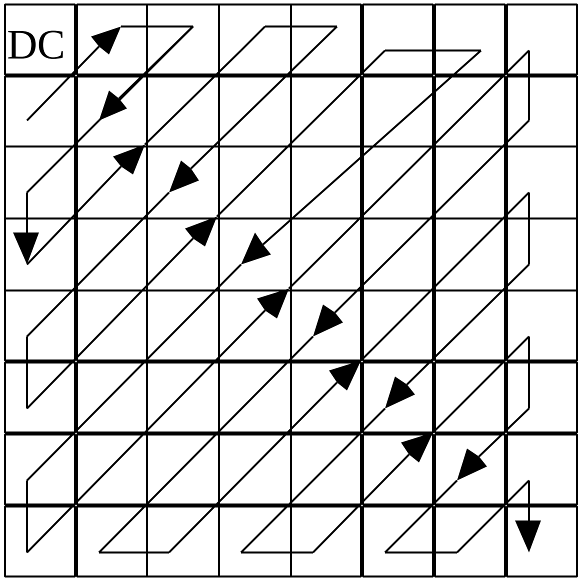
The JPEG committee has listed the following luminance quantization table as an example in Annex K (informative) of the IS. It is obtained by measuring the DCT coefficient “visibility threshold” using CCIR-601 images and display, at a distance of six picture-heights away.

$Q_L(u,v) =$

16	11	10	16	24	40	51	61
12	12	14	19	26	58	60	55
14	13	16	24	40	57	69	56
14	17	22	29	51	87	80	62
18	22	37	56	68	109	103	77
24	35	55	64	81	104	113	92
49	64	78	87	103	121	120	101
72	92	95	98	112	100	103	99

JPEG DCT Example

- After Quantization, the DCT is separated into a DC coefficient and AC coefficients, which are reordered into a 1-D format using a zigzag pattern in order to create long runs of zero-valued coefficients.
  - The DC coefficient is directly correlated to the mean of the 8-by-8 block (upper-left corner).
    - All DC coefficients are combined into a separate bit stream.
  - The AC coefficients are the values of the cosine basis functions (all other values).



JPEG DCT Example

DC	AC								
15	0	-2	-1	-1	-1	0	0	-1	EOB

- The DC coefficient is encoded using Huffman encoded 1D-DPCM
- The AC coefficients are encoded using Huffman coding on magnitude/runlength pairs (magnitude of a nonzero AC coefficient plus runlength of zero-valued AC coefficients that precede it).
- The end-of-block (EOB) symbol indicates that all remaining coefficients in the zigzag scan are zero. This allows many coefficients to be encoded with only a single symbol.

JPEG DCT Example

Dequantized DCT Coefficients

240	0	-10	0	0	0	0	0
-24	-12	0	0	0	0	0	0
-14	-13	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

Reconstructed Image

144	146	149	152	154	156	156	156
148	150	152	154	156	156	156	156
155	156	157	158	158	157	156	155
160	161	161	162	161	159	157	155
163	163	164	163	162	160	158	156
163	163	164	164	162	160	158	157
160	161	162	162	162	161	159	158
158	159	161	161	162	161	159	158

Error Image

-5	-2	0	1	1	-1	-1	-1
-4	1	1	2	3	0	0	0
-5	-1	3	5	0	-1	0	1
-1	0	1	-2	-1	0	2	4
-4	-3	-3	-1	0	-5	-3	-1
-2	-2	-3	-3	-2	-3	-1	0
2	1	-1	1	0	-4	-2	-1
4	3	0	0	1	-3	-1	0

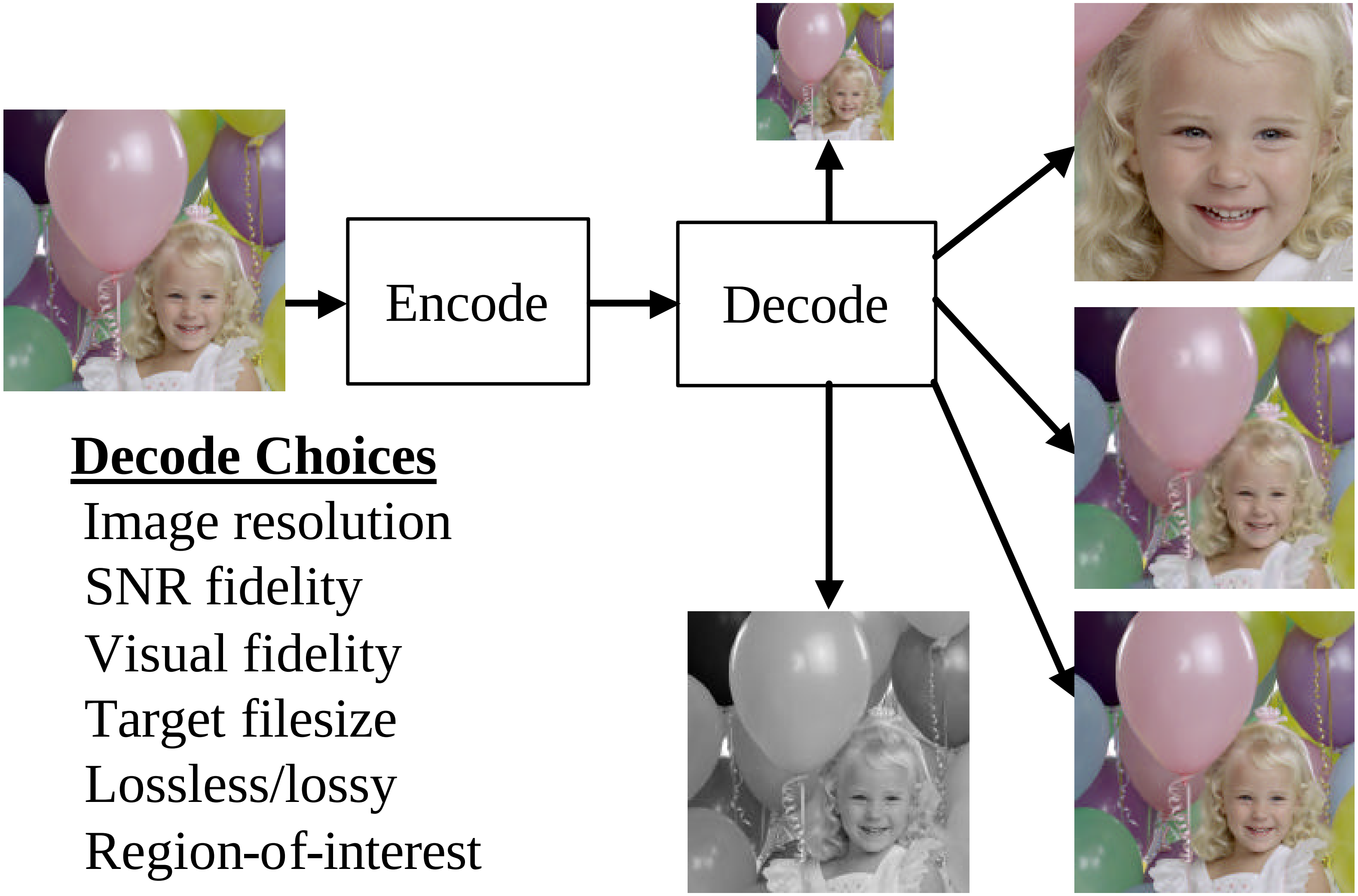
RMSE= 2.26

The Emerging JPEG2000 Standard

JPEG-DCT Pros and Cons

- Advantages
  - Memory efficient
  - Low complexity
  - Compression efficiency
  - Visual model utilization
  - Robustness
- Disadvantages
  - Single resolution
  - Single quality
  - No target bit rate
  - No lossless capability
  - No tiling
  - No region of interest
  - Blocking artifacts
  - Poor error resilience

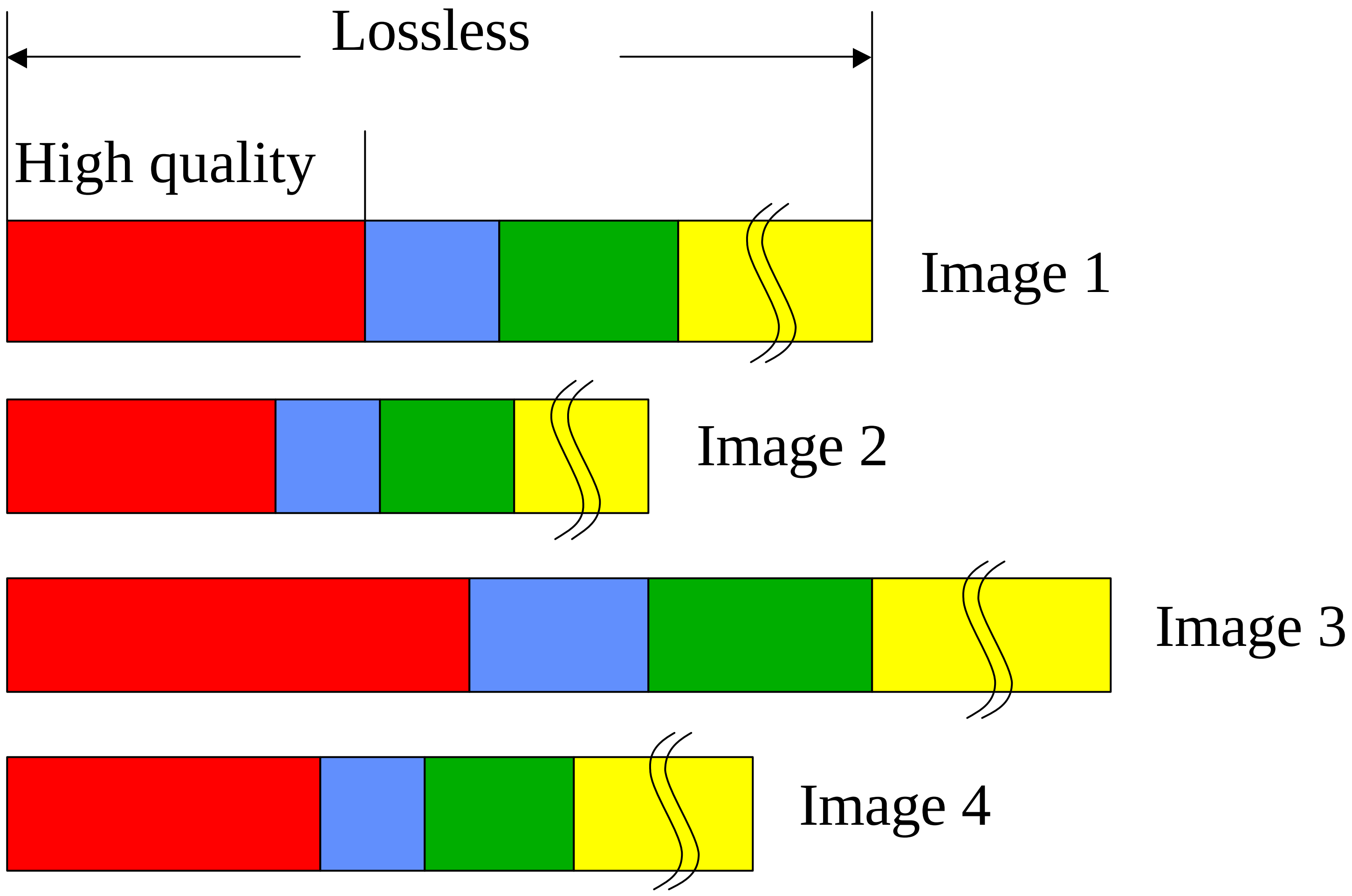
JPEG2000 Compression Paradigm



Decode Choices

Image resolution
SNR fidelity
Visual fidelity
Target filesize
Lossless/lossy
Region-of-interest
Tiles

Embedded Bitstream Application



JPEG2000 Objectives

- Advanced standardized image coding system to serve applications into the next millenium
- Address areas where current standards fail to produce the best quality or performance
- Provide capabilities to markets that currently do not use compression
- Provide an open system approach to imaging applications

JPEG2000: Requirements and Profiles

- Internet applications (World Wide Web imagery)
  - Progressive in quality and resolution, fast decode
- Mobile applications
  - Error resilience, low power, progressive decoding
- Digital photography
  - Low complexity, compression efficiency
- Hardcopy color facsimile, printing and scanning
  - Compression efficiency, strip or tile processing
- Digital library/archive applications
  - Metadata, content management
- Remote sensing
  - Multiple components, fast encoding, region of interest
- Medical applications
  - Region of interest coding, lossy to lossless

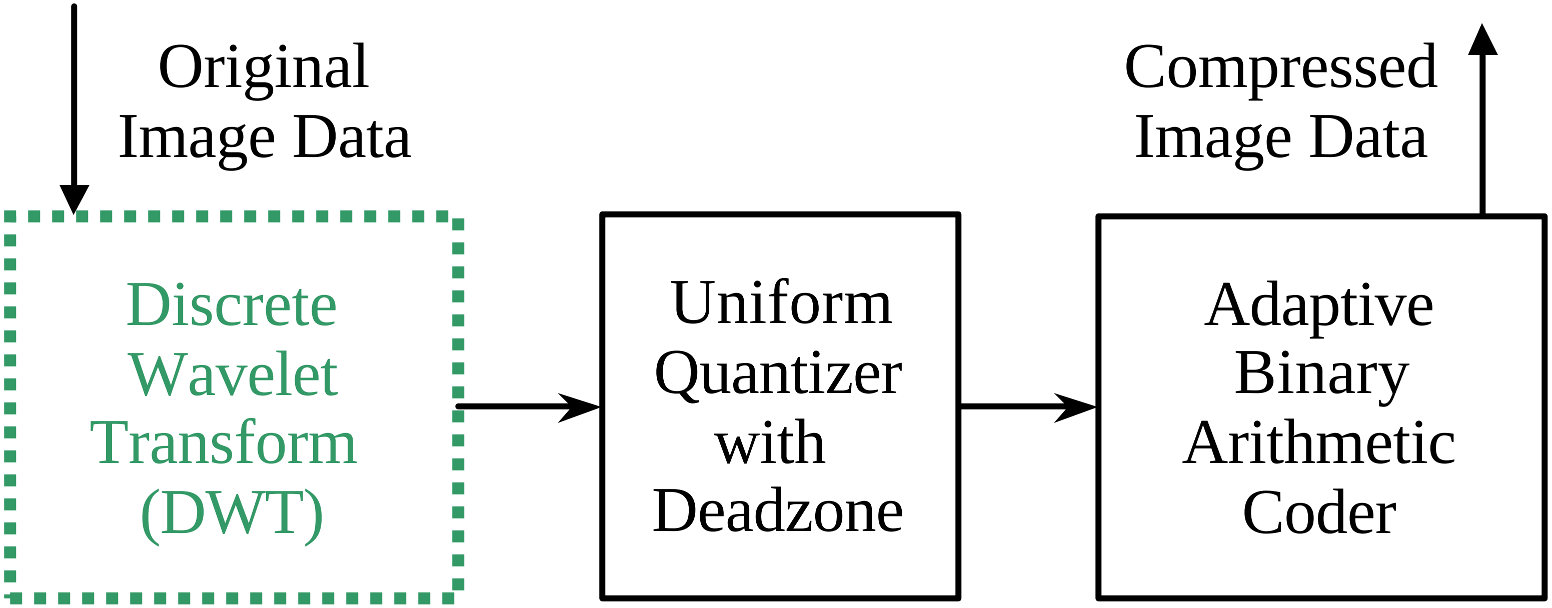
JPEG2000 Features

- Improved compression efficiency (estimated 5-30% depending on the image size and bit rate)
- Lossy to lossless
- Multiple resolution
- Embedded bit stream (progressive decoding)
- Region of interest coding (ROI)
- Error resilience
- Bit stream syntax
- File format

JPEG2000 Compression Standard

- The standard only specifies a decoder and a bitstream syntax and is issued in several parts:
  - **Part I:** specifies the minimum compliant decoder (e.g., a decoder that is expected to satisfy 80% of applications); International Standard (IS) is expected 12/2000.
  - **Part II:** Describes optional, value added extensions; IS is expected in 12/2001.
  - **Other parts** include: Motion JPEG2000 (Part III); Conformance Testing (Part IV); reference software in JAVA and C (Part V); file format for compound images (Part VI); and Technical Report outlining guidelines for minimum support of Part I (Part VII).

JPEG2000 Part I Encoder



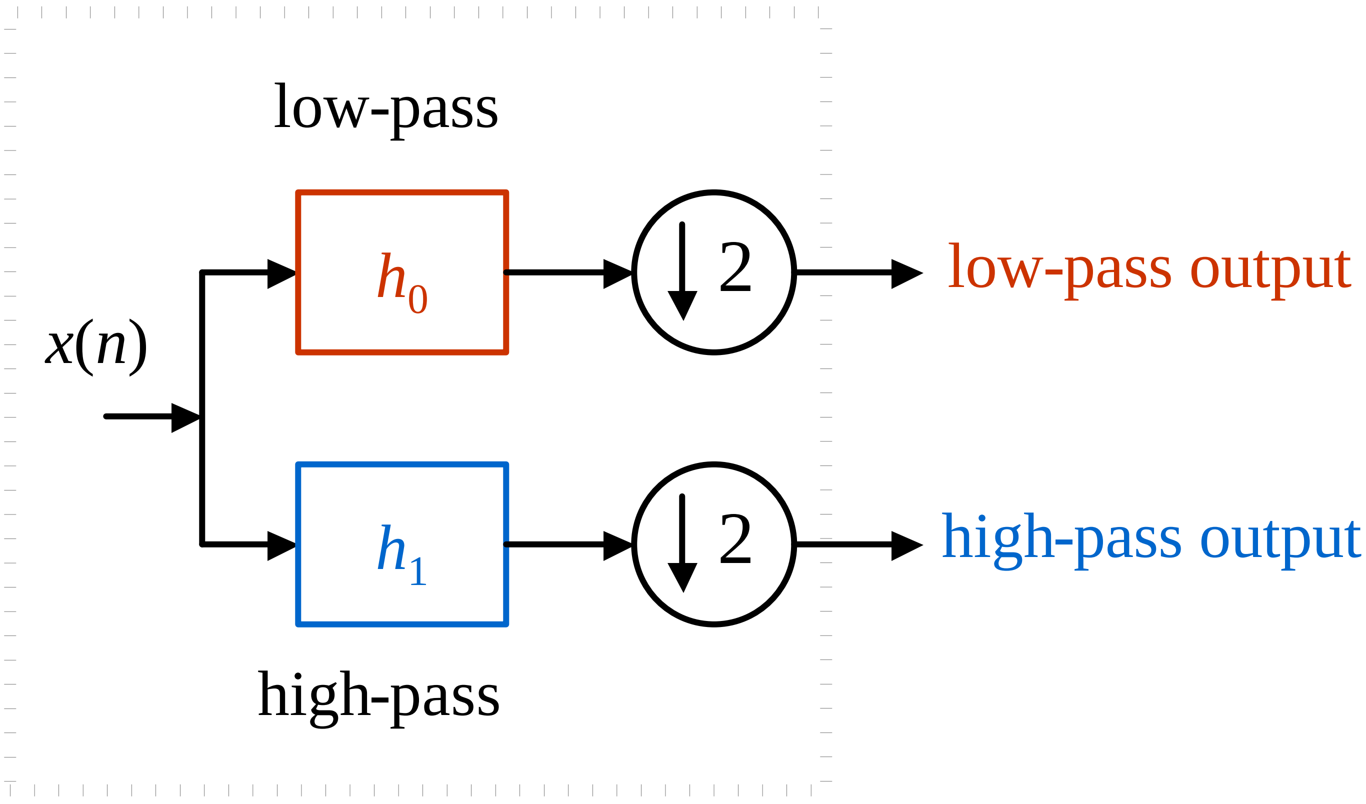
Benefits of DWT

- Multiple resolution representation
- Lossless representation with integer filters
- Better decorrelation than DCT, resulting in higher compression efficiency
- Use of visual models
  - DWT provides a frequency band decomposition of the image where each subband can be quantized according to its visual importance (similar to the quantization table specification in JPEG-DCT)

JPEG2000 DWT Choices

- JPEG-2000 Part I only allows successive powers of two splitting of the LL band and the use of two DWT filters:
  - The integer (5,3) filter provides fast implementation (faster than DCT) and lossless capability, but at the expense of some loss in coding efficiency.
  - The Daubechies (9,7) floating-point filter that provides superior coding efficiency. The analysis filters are normalized to a DC gain of one and a Nyquist gain of 2.
- Part II allows for arbitrary size filters (user-specified in the header), arbitrary wavelet decomposition trees, and different filters in the horizontal vs. vertical directions.

The 1-D Two-Band DWT



Analysis filter bank

Example of Analysis Filter Bank

- 1-D signal:

  …100 100 100 100 200 200 200 200...

- Low-pass filter h_0: (-1 2 6 2 -1)/8

- High-pass filter h_1: (-1 2 -1)/2

- Before downsampling:

  … 100 $\underline{100}$ 87.5 $\underline{112.5}$ 187.5 $\underline{212.5}$ 200 $\underline{200}$...

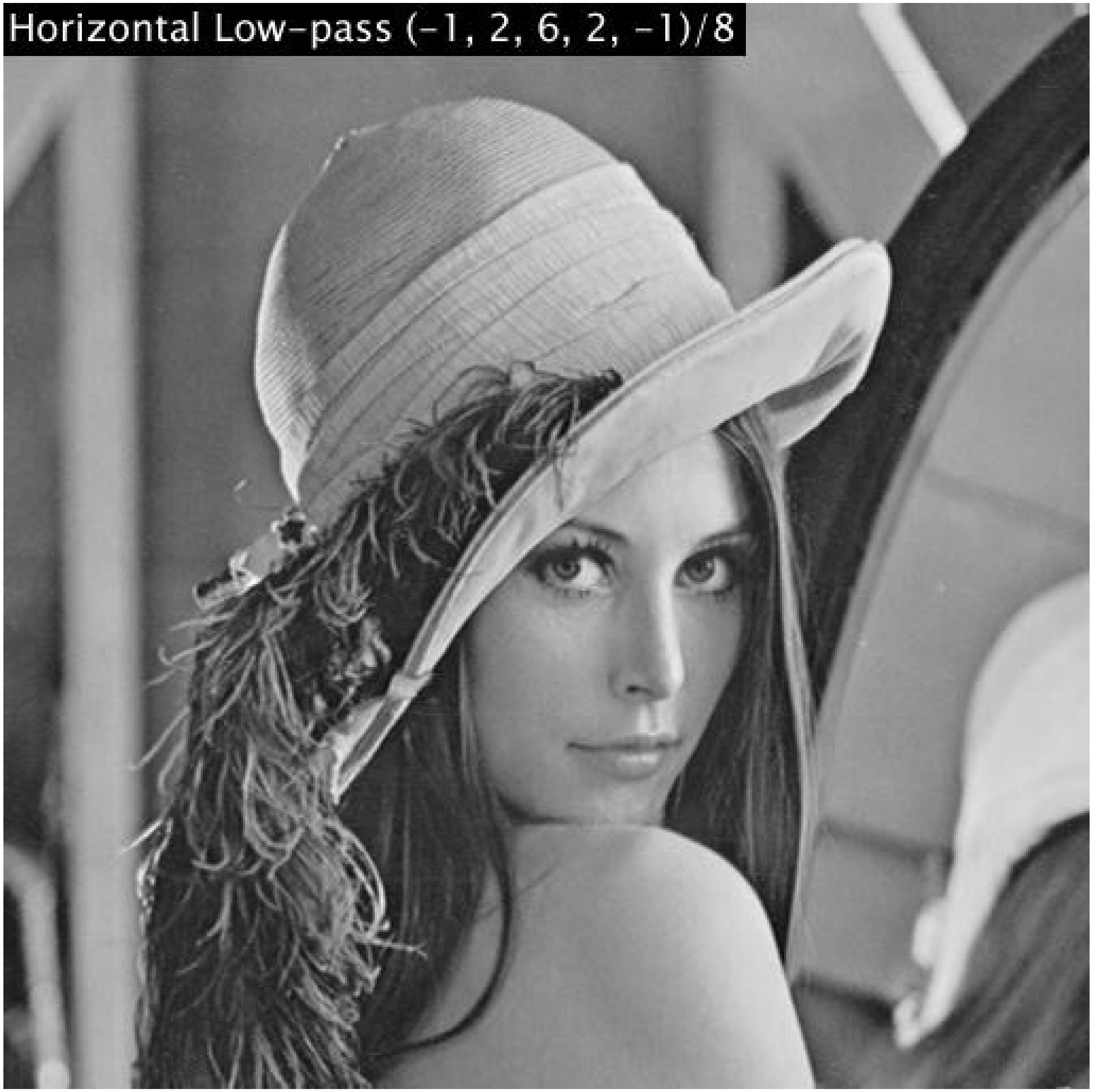
  … $\underline{0}$ 0 $\underline{0}$ -50 $\underline{50}$ 0 $\underline{0}$ 0…

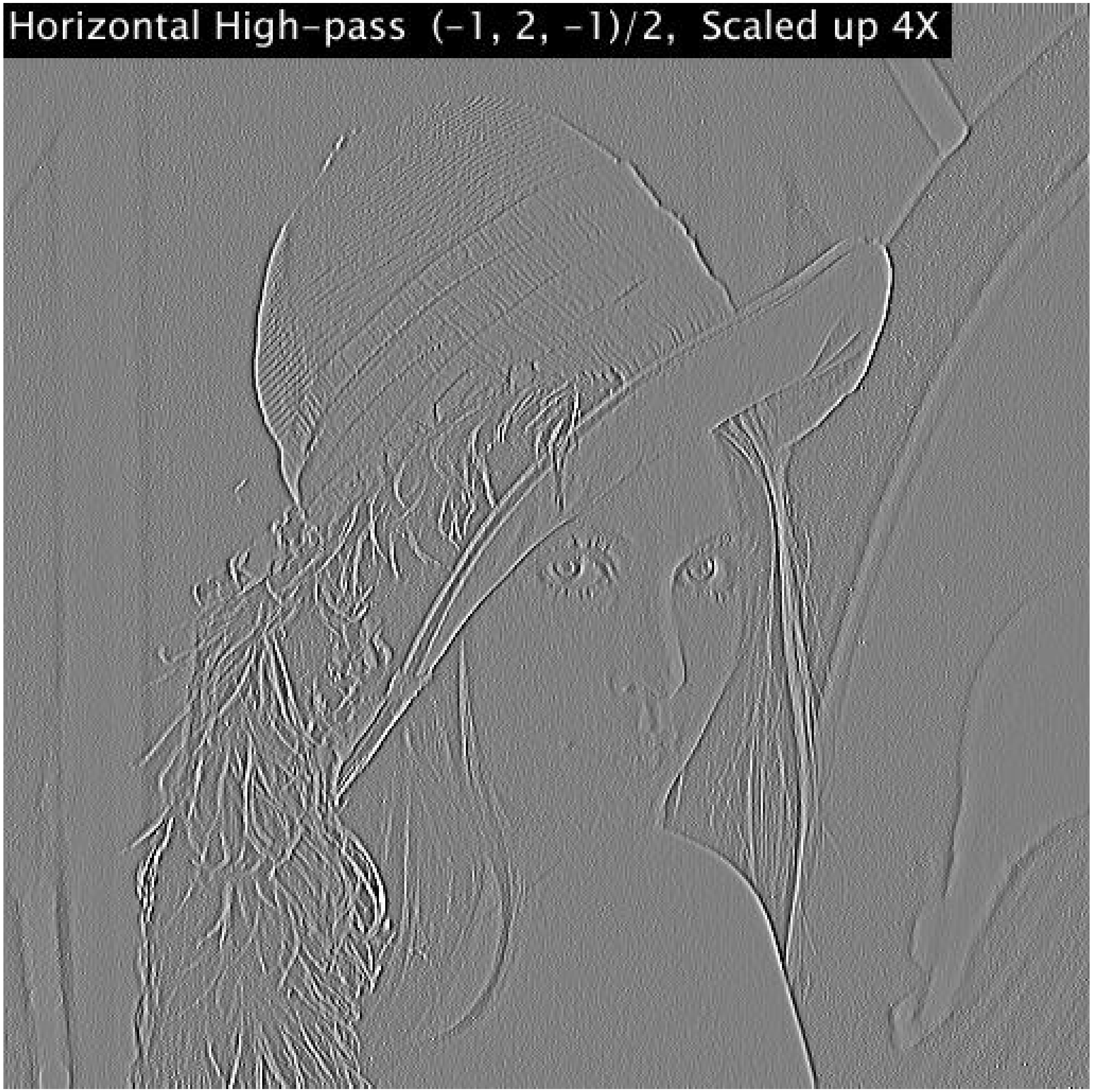
- After downsampling:

  … 100 112.5 212.5 200...

  … 0 0 50 0 …

Horizontal Low-pass (-1, 2, 6, 2, -1)/8





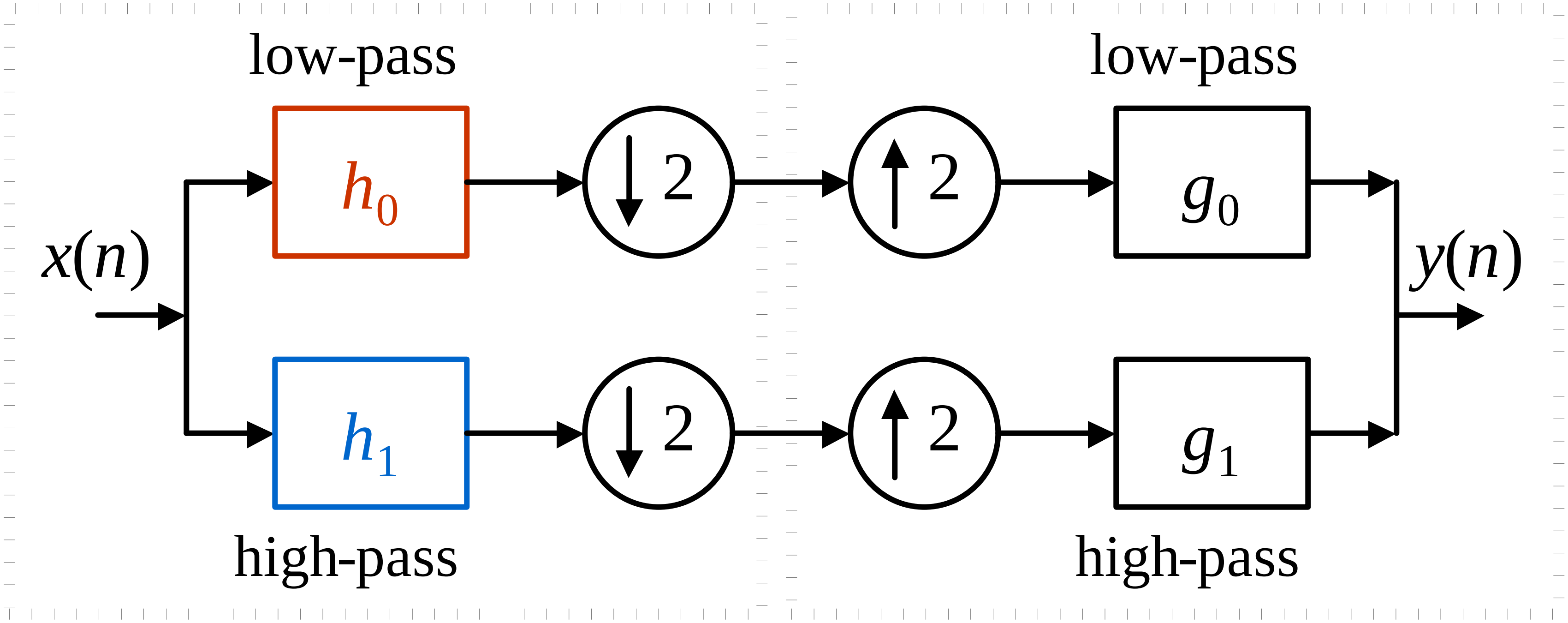
Horizontal High-pass (-1, 2, -1)/2, Scaled up 4X

Inverse DWT

- During the inverse DWT, each subband is interpolated by a factor of two by inserting zeros between samples and then filtering each resulting sequence with the corresponding low-pass, g_0, or high-pass, g_1, **synthesis filters**.
- The filtered sequences are added together to form an approximation to the original signal.

…	**0**	100	**0**	112.5	**0**	212.5	**0**	200…
…	0	**0**	0	**0**	50	**0**	0	**0**…
…	100	100	100	100	200	200	200	200...

The 1-D Two-Band DWT



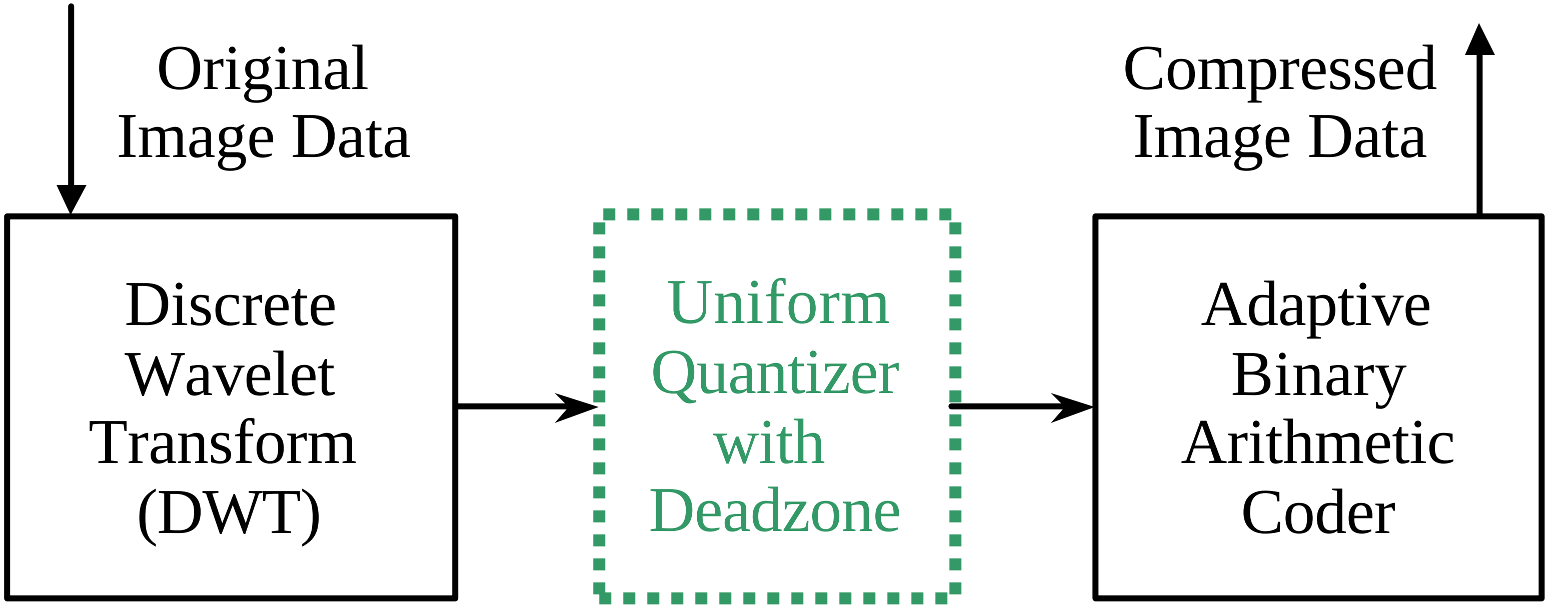
Analysis filter bank

Synthesis filter bank

Ideally, it is desired to choose the analysis filter banks (h_0 and h_1), and the synthesis filter banks (g_0 and g_1), in such a way so as to make the overall distortion zero, i.e., $x(n) = y(n)$. This is called the **perfect reconstruction** property.

NIMA Systems Engineering Services

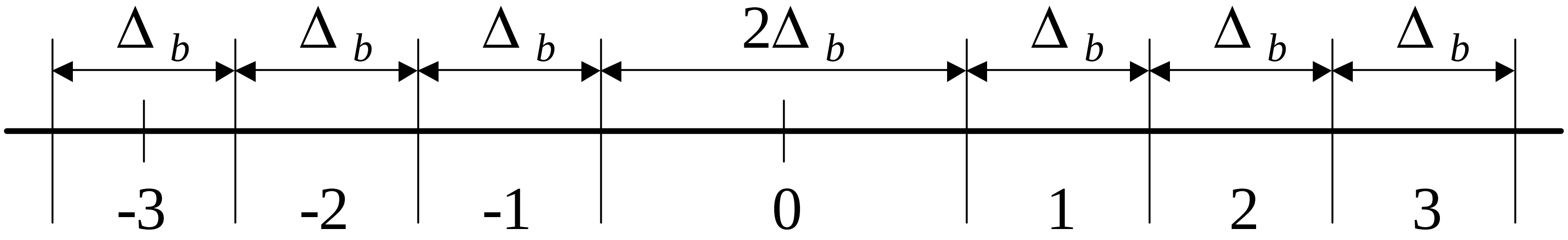
JPEG2000 Part I Encoder


Original Image Data
Discrete Wavelet Transform (DWT)
Uniform Quantizer with Deadzone
Adaptive Binary Arithmetic Coder
Compressed Image Data

Quantization step size can vary from one subband to another according to visual models (similar to JPEG Q-table specification).

Quantization in Part I

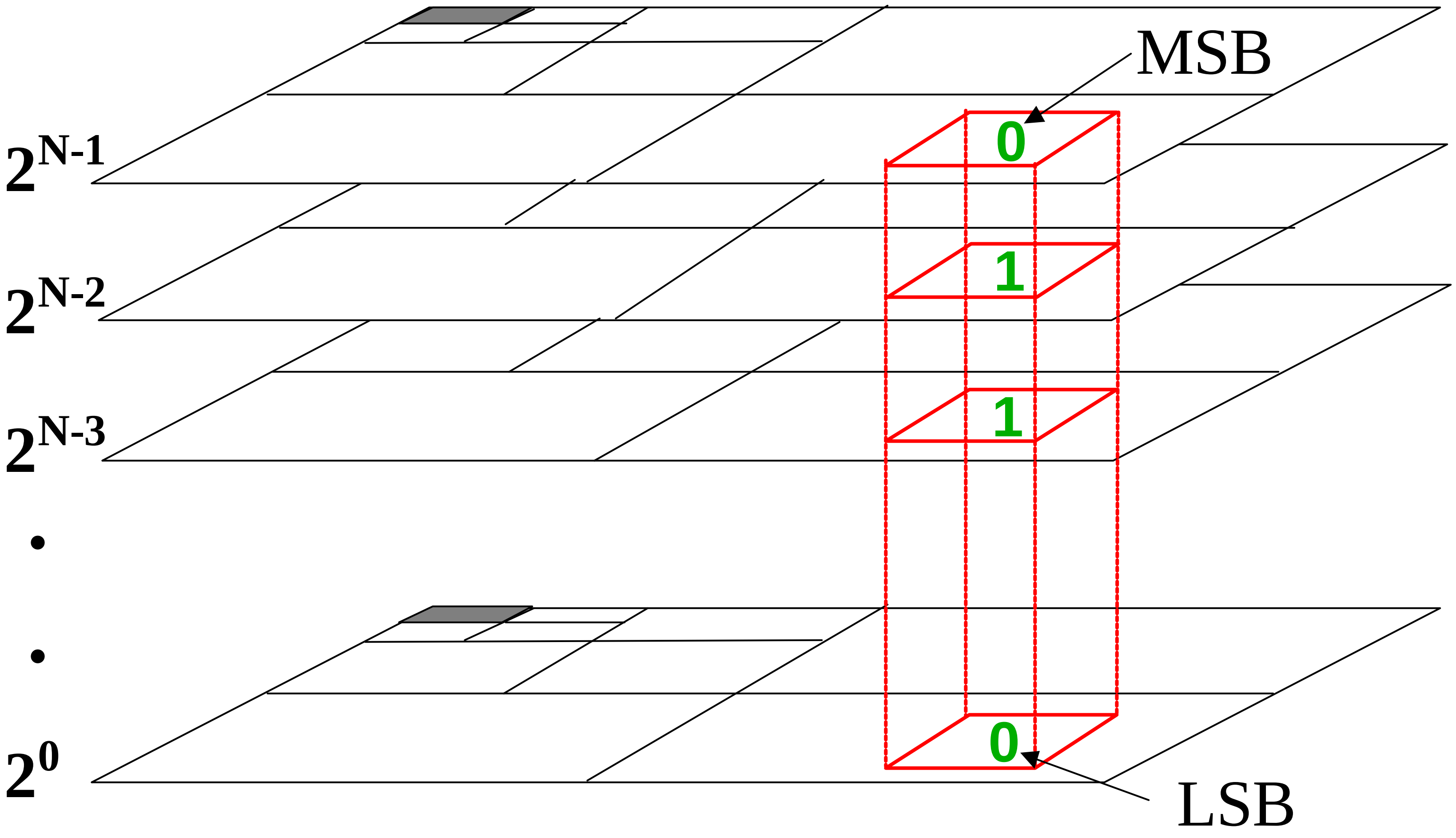
- Uniform quantization with deadzone is used to quantize all the wavelet coefficients.
- For each subband *b*, a basic quantizer step size Δ_b is selected by the user and is used to quantize all the coefficients in that subband.
- The choice of the quantizer step size for each subband can be based on visual models and is likened to the q-table specification in the JPEG DCT.



Embedded Quantization in Part I

- Unlike JPEG Baseline, where the resulting quantizer index q is encoded as a single symbol, in JPEG2000 it is encoded one bit at a time, starting from the MSB and proceeding to the LSB.
- During this progressive encoding, the quantized wavelet coefficient is called **insignificant** if the quantizer index q is still zero. Once the first nonzero bit is encoded, the coefficient becomes **significant** and its sign is encoded.
- If the p least significant bits of the quantizer index still remain to be encoded, the reconstructed sample at that stage is identical to the one obtained by using a UTQ with deadzone with a step size of $\Delta_b 2^p$.

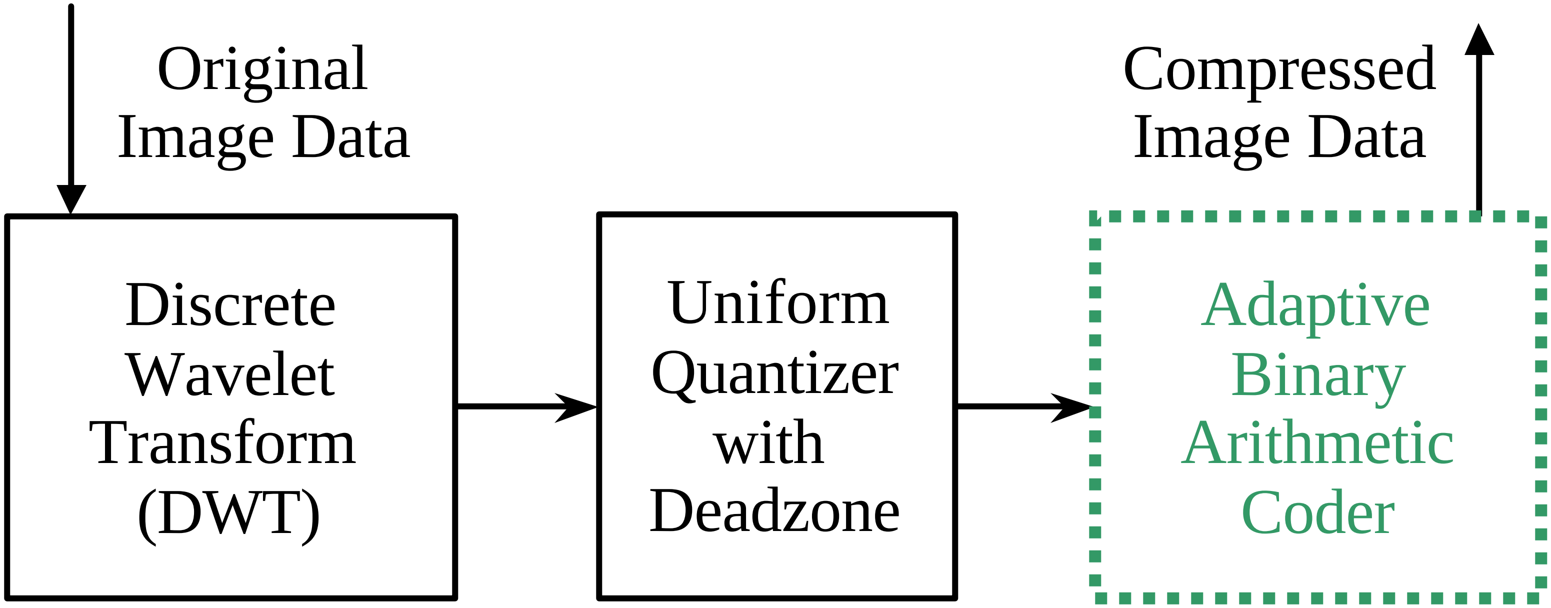
Embedded Quantization by Bit-Plane Coding



Embedded Quantization Example

- Wavelet coefficient = 83
  - Quantizer step size = 3
  - Quantizer index = $\lfloor 83/3 \rfloor$ = 27
  - Quantizer index in binary: 00011011
  - Decoded index after 6 BP’s: 000110 = 6
  - Step size with 2 BP’s remaining = 12
  - Quantizer index with step size of 12 = 6
  - Dequantized value = (6 + 0.5) x 12 = 78

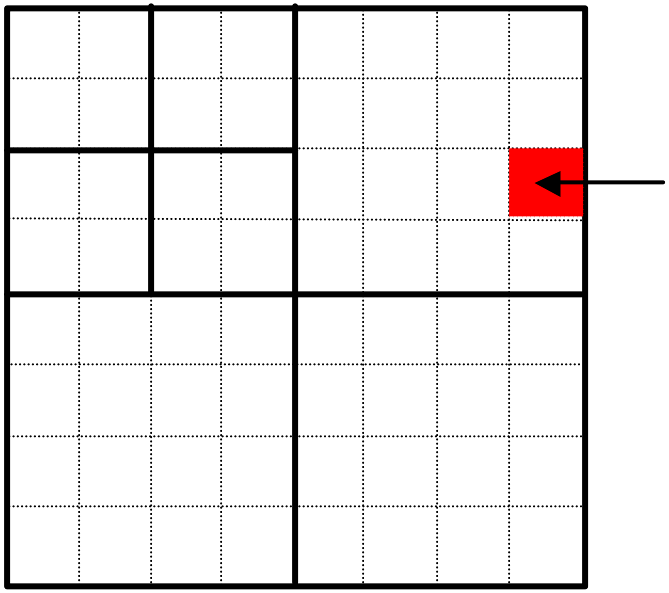
JPEG2000 Part I Encoder


Original Image Data
Compressed Image Data
Discrete Wavelet Transform (DWT)
Uniform Quantizer with Deadzone
Adaptive Binary Arithmetic Coder

Context-based adaptive binary arithmetic coding is used in JPEG2000 to efficiently compress each individual bit plane.

JPEG2000 Entropy Coder

- Each bit plane is further broken down into blocks (e.g., 64 x 64). The blocks are coded independently (i.e., the bit stream for each block can be decoded independent of other data) using three coding passes. The coding progresses from the most significant bit-plane to the least significant bit-plane.



A coding block of a bit plane of a subband

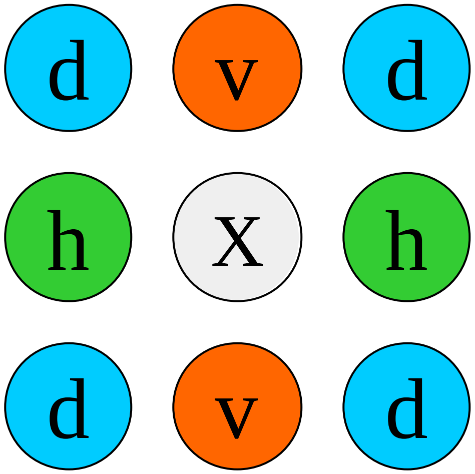
JPEG2000 Entropy Coder

- The binary value of a sample in a block of a bit plane of a subband is coded as a binary symbol with the JBIG2 **MQ-Coder** that is a context-based adaptive arithmetic coder.
- Each bit-plane of each block of a subband is encoded in **three sub bit plane passes** instead of a single pass. The bitstream can be truncated at the end of each pass. This allows for:
  - Optimal embedding, so that the information that results in the most reduction in distortion for the least increase in file size is encoded first.
  - A larger number of bit-stream truncation points to achieve finer SNR scalability.

Significance Propagation Pass

- The first pass in a new bit plane is called the **significance propagation pass**. A symbol is encoded if it is insignificant but at least one of its eight-connected neighbors is significant as determined from the previous bit plane and the current bit plane based on coded information up to that point. These locations have the highest probability of becoming significant).

- The probability of the binary value at a given location of a bit-plane of a block of a subband is modeled by a context formed from the significance values of its neighbors.



Refinement and Clean-up Passes

- **Refinement** (REF): Next, the significant coefficients are refined by their bit representation in the current bit-plane.
- **Clean-up**: Finally, all the remaining coefficients in the bit-plane are encoded. (*Note*: the first pass of the MSB bit-plane of a subband is always a clean-up pass).
- The coding for the first and third passes are identical, except for the run coding that is employed in the third pass.
- The maximum number of contexts used in any pass is no more than nine, thus allowing for extremely rapid probability adaptation that decreases the cost of independently coded segments.

NIMA
Systems Engineering Services

Kodak

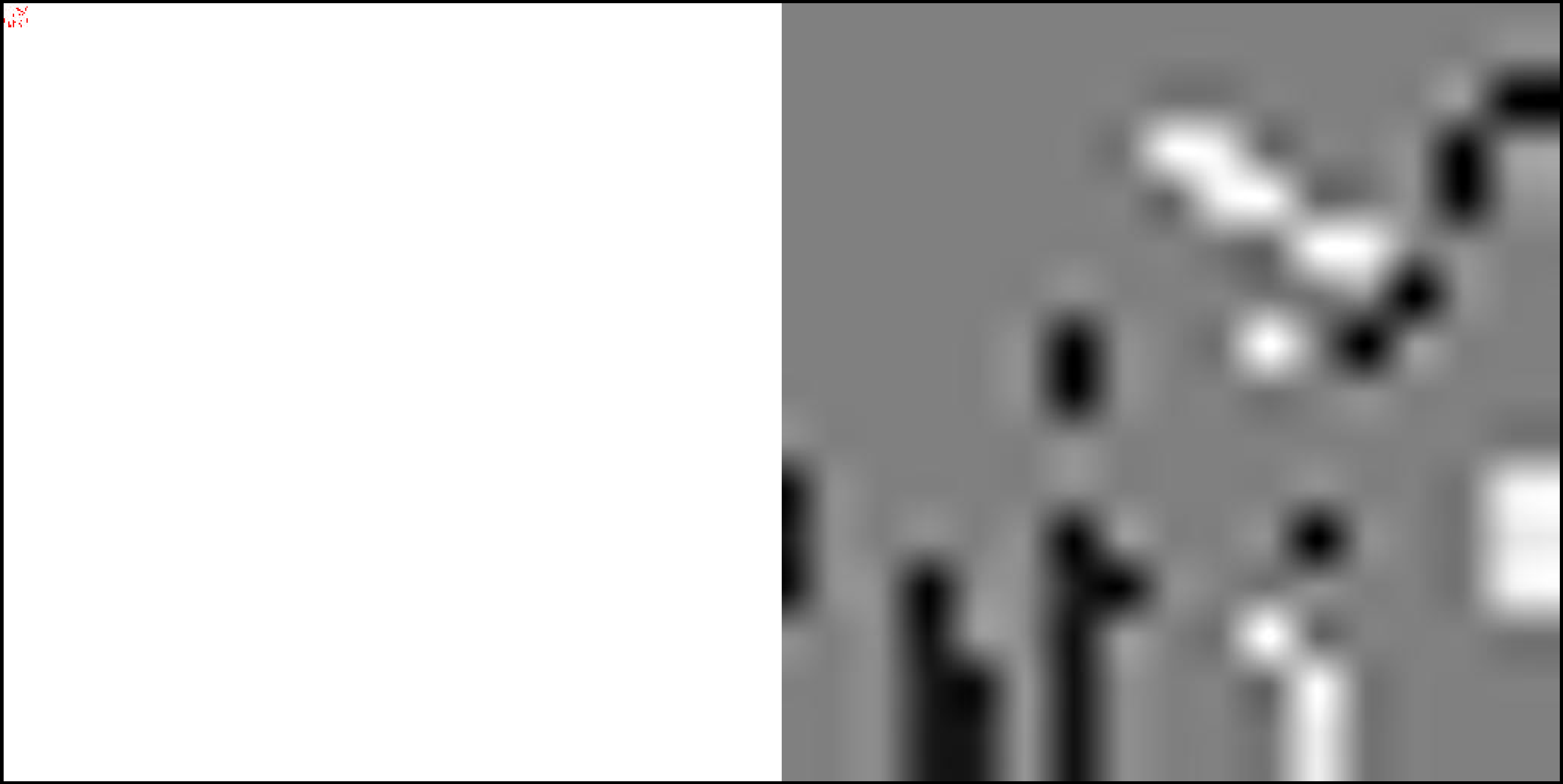
Sig. Prop.	=	0
Refine	=	0
Cleanup	=	21
Total Bytes		21

Bit plane 1

Compression ratio = 12483 : 1

RMSE = 39.69 PSNR = 16.16 db

% refined = 0 % insig. = 99.99



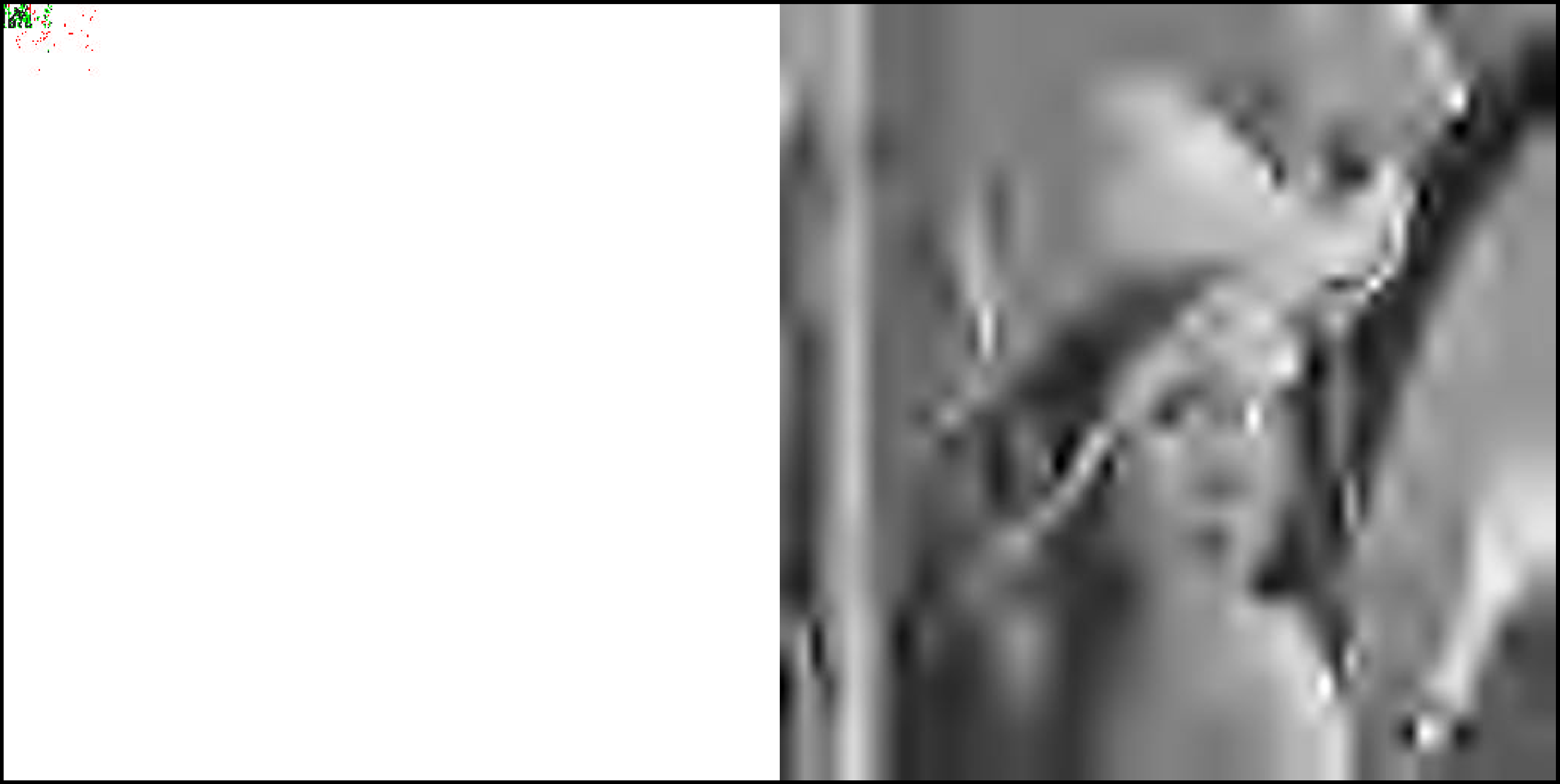
Sig. Prop.	=	38
Refine	=	13
Cleanup	=	57
Total Bytes		108

Bit plane **3**

Compression ratio = 1533 : 1

RMSE = 21.59 PSNR = 21.45 db

% refined = 0.05 % insig. = 99.89



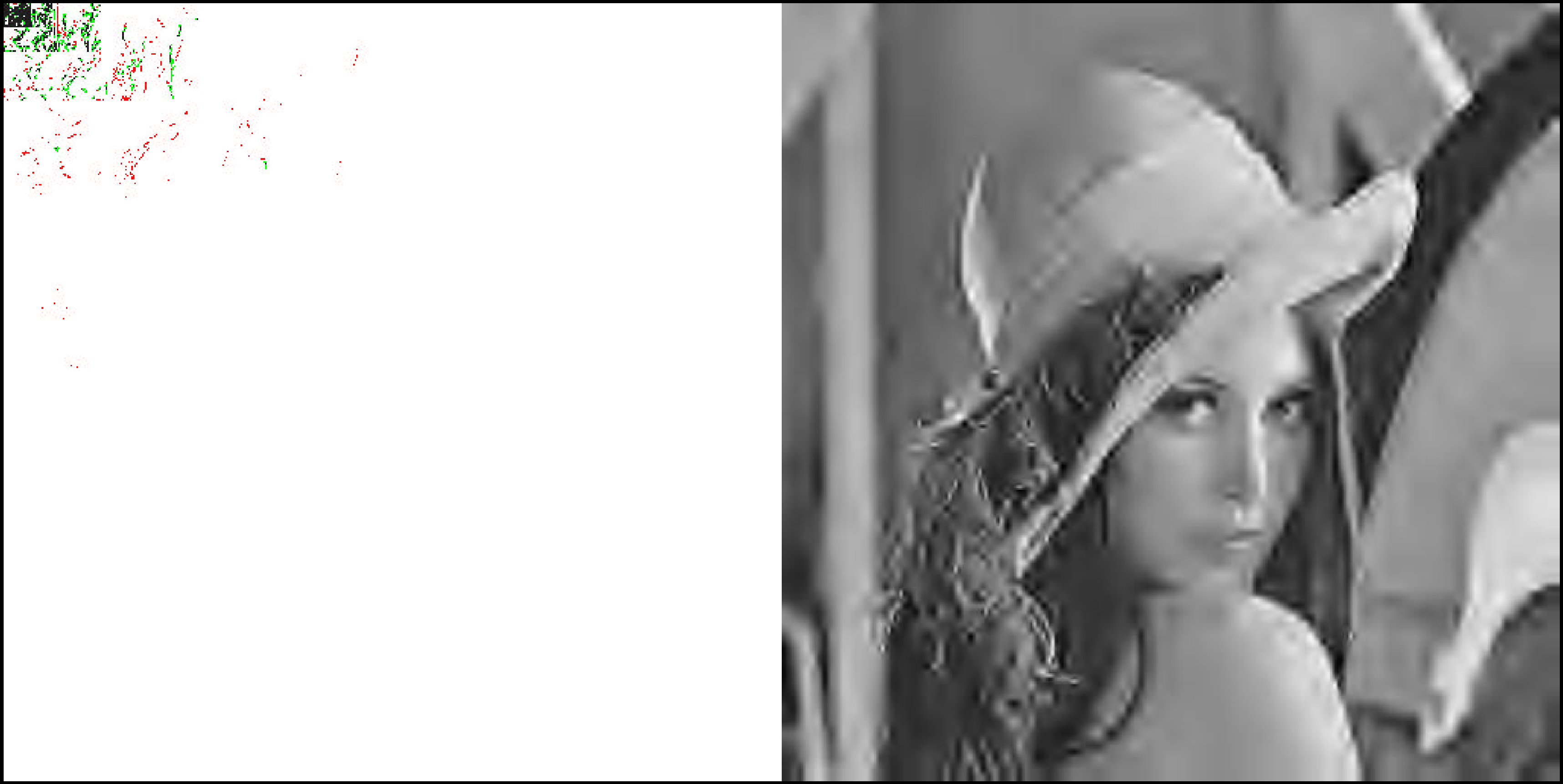
Sig. Prop.	=	224
Refine	=	73
Cleanup	=	383
Total Bytes		680

Bit plane **5**

Compression ratio = 233 : 1

RMSE = 12.11 PSNR = 26.47 db

% refined = 0.23 % insig. = 99.43



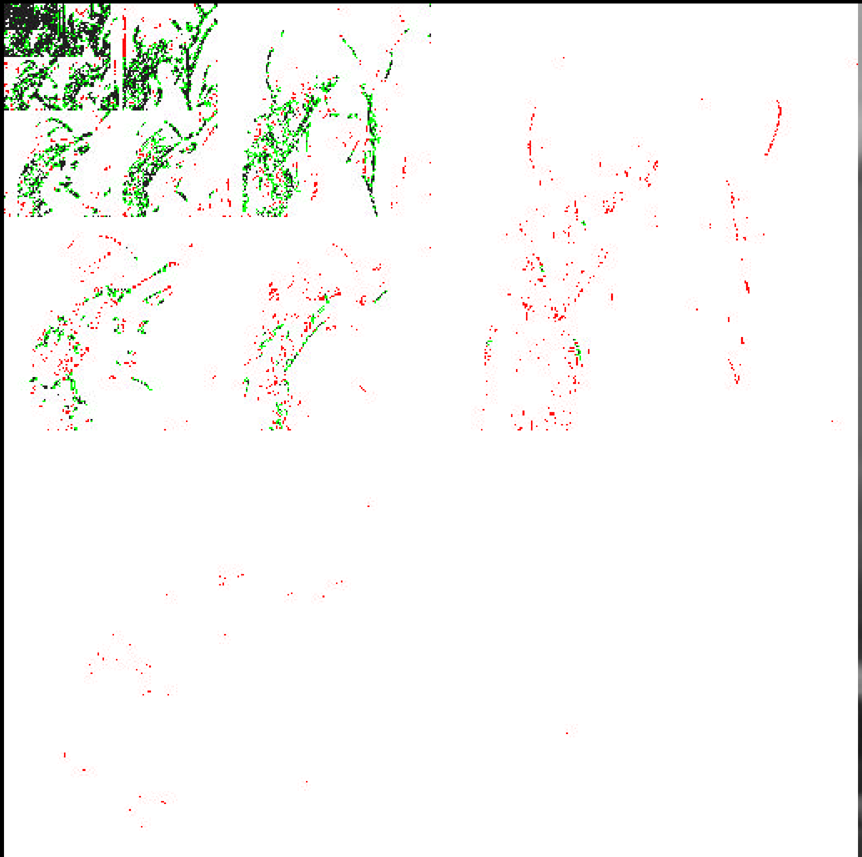
Sig. Prop.	=	1243
Refine	=	418
Cleanup	=	1349
Total Bytes		3010

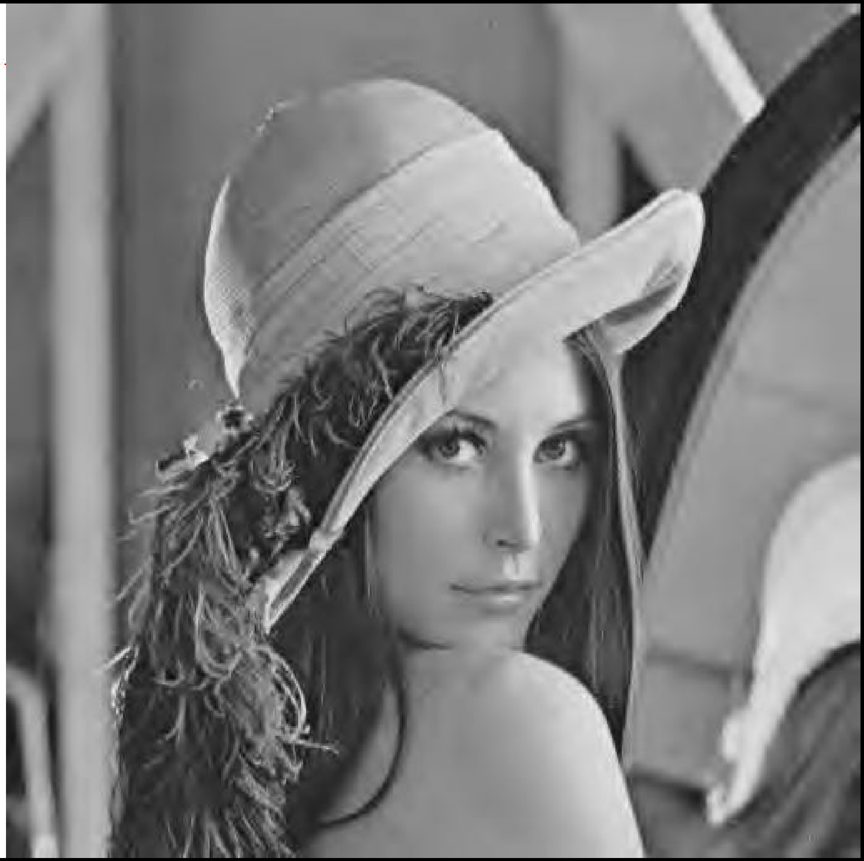
Bit plane **7**

Compression ratio = 47 : 1

RMSE = 6.02   PSNR = 32.54 db

% refined = 1.32   % insig. = 97.09





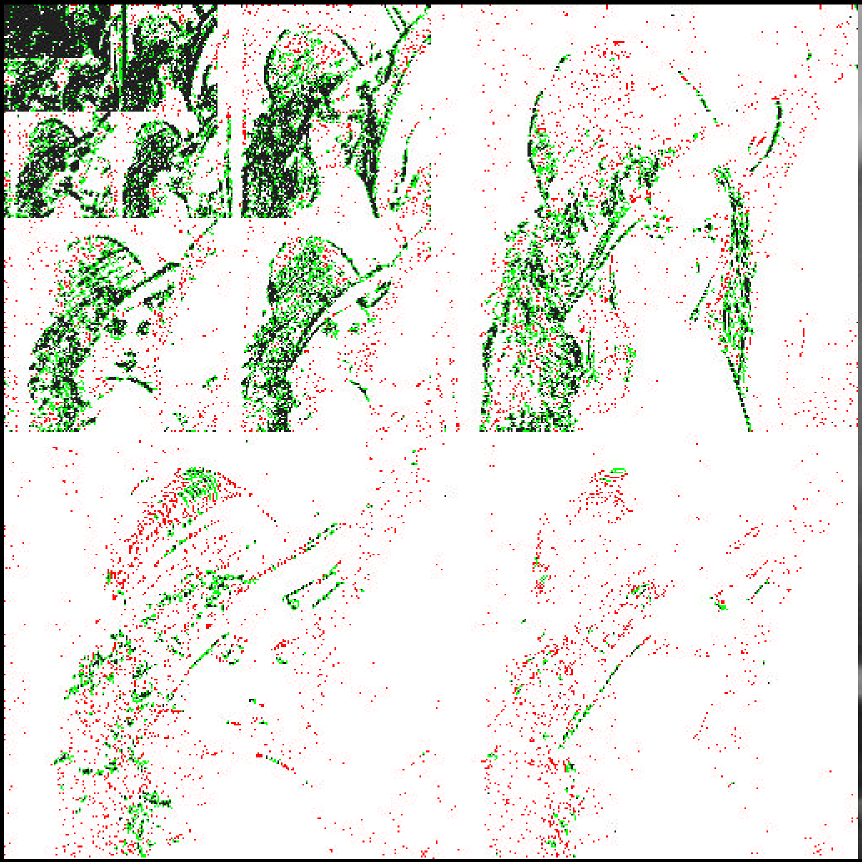
Sig. Prop.	=	4593
Refine	=	1925
Cleanup	=	5465
Total Bytes		11983

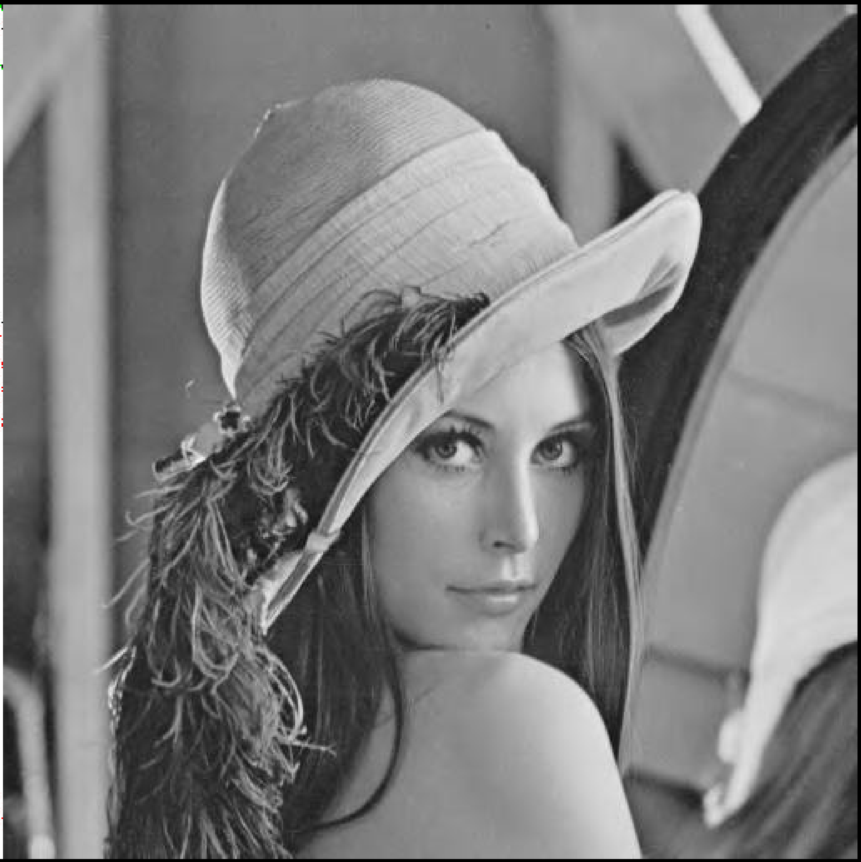
Bit plane **9**

Compression ratio = 11.2 : 1

RMSE = 2.90 PSNR = 38.87 db

% refined = 6.01 % insig. = 87.66





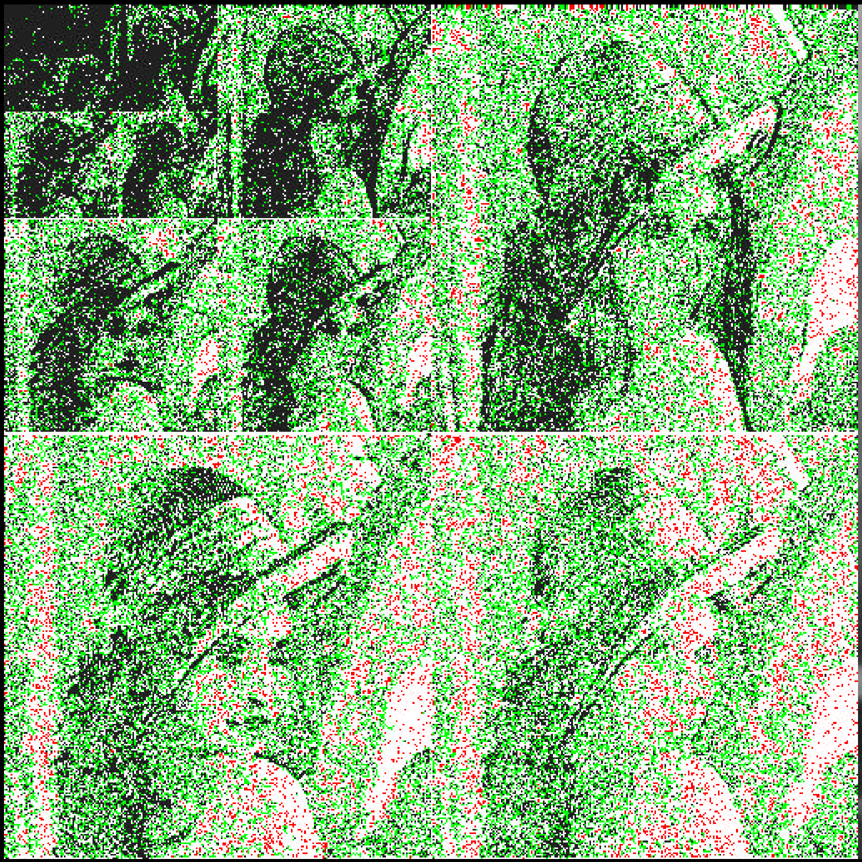
Sig. Prop.	=	25421
Refine	=	8808
Cleanup	=	5438
Total Bytes		39667

Bit plane **11**

Compression ratio = 2.90 : 1

RMSE = 0.90 PSNR = 49.00 db

% refined = 28.12 % insig. = 46.80



Wavelet Bit Plane Compression Table

BP	JPEG-2K Bytes[†]	JPEG-2K PSNR-dB	JPEG-2K RMSE	JPEG Bytes[‡]	JPEG PSNR	JPEG RMSE
1	21	16.16	39.69			
2	63	18.85	29.11			
3	171	21.45	21.59			
4	442	23.74	16.58			
5	1122	26.47	12.11			
6	2601	29.39	8.65			
7	5821[*]	32.54	6.02	5804[*]	29.77	8.28
8	11680[*]	35.70	4.18	11696[*]	33.33	5.49
9	23716[*]	38.87	2.90	23932[*]	36.50	3.81
10	51164[*]	43.12	1.78	51636[*]	40.16	2.50
11	90855[*]	49.00	0.90	91216[*]	44.12	1.59

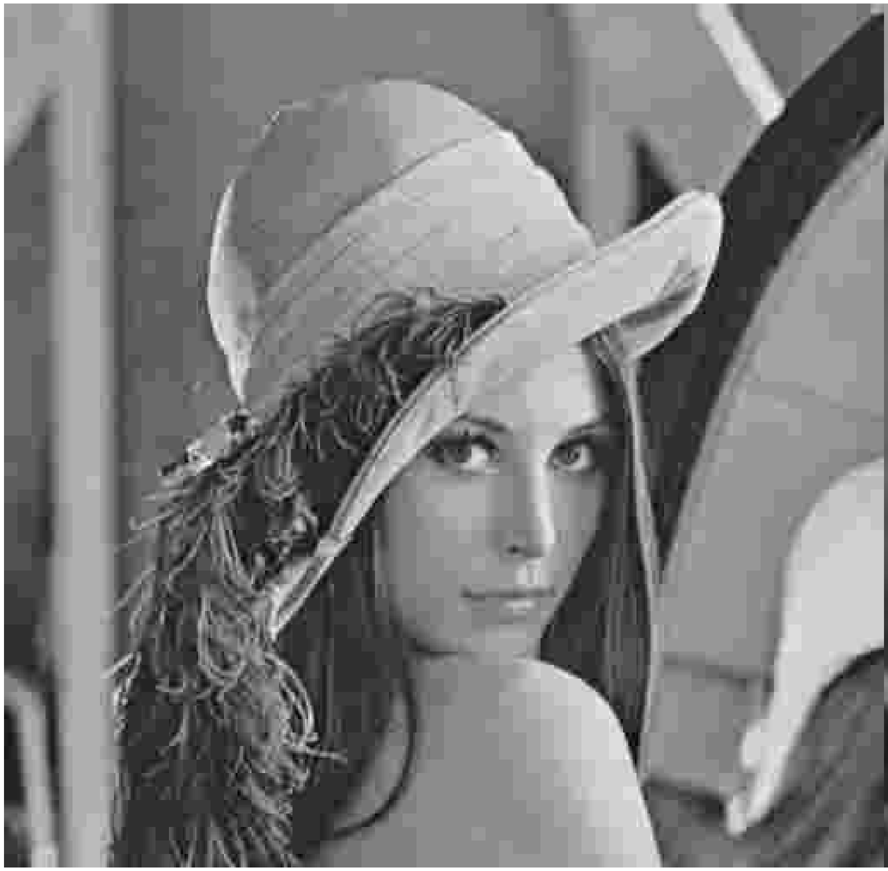
[†] 6-level decomposition with (9,7) filter

[‡] IJG code with default q-table

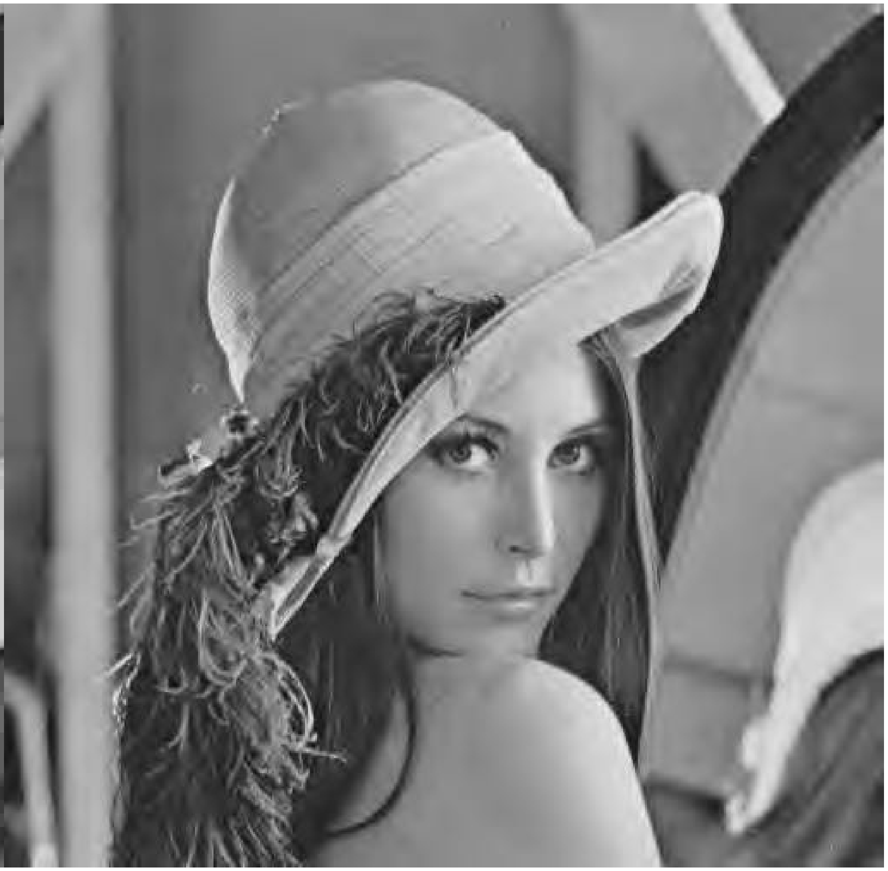
[*] Filesize includes the header

Comparison of JPEG and JPEG-2000

JPEG, PSNR = 29.77 db
Filesize = 5804 Bytes



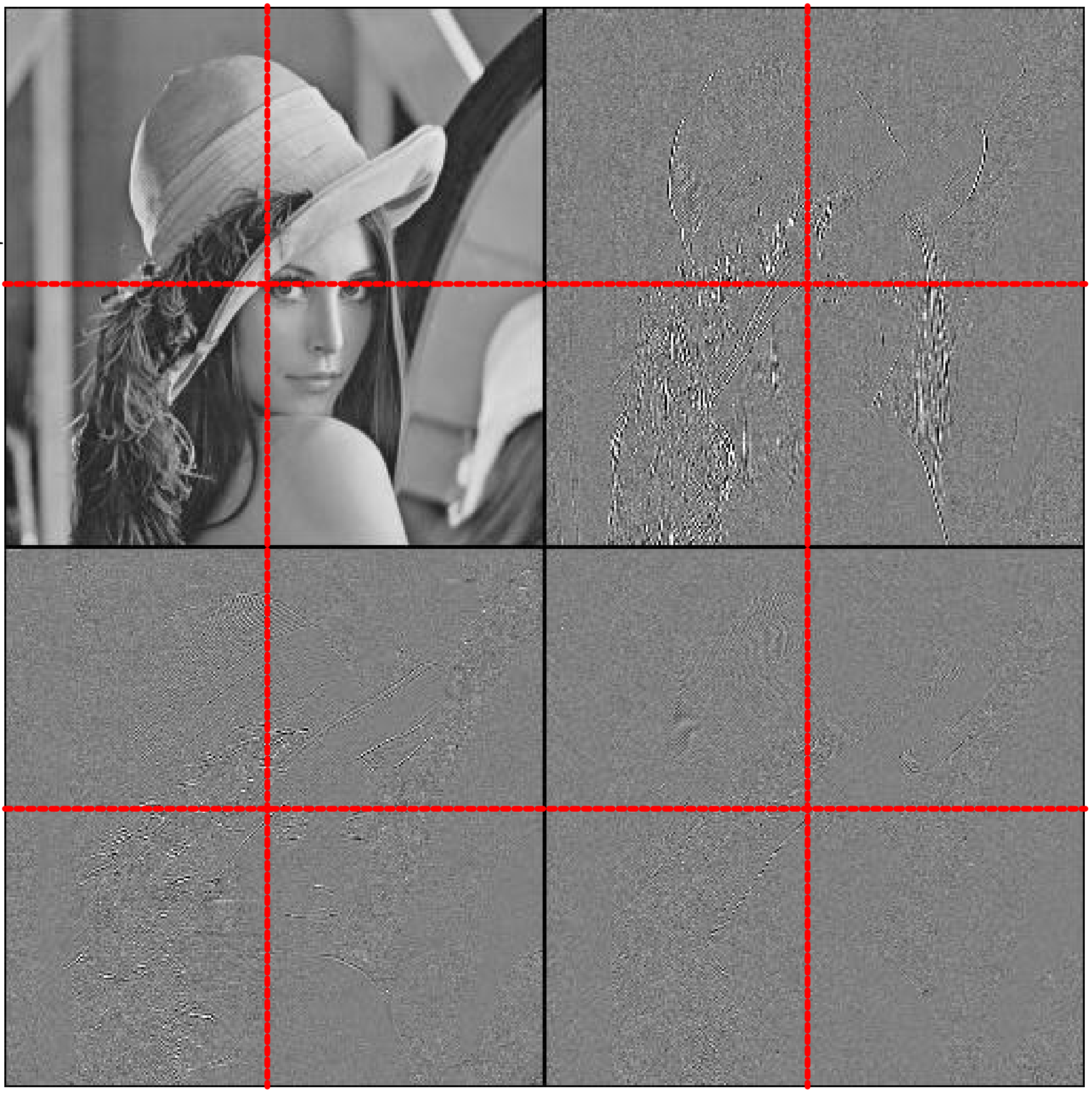
JPEG-2K, PSNR = 32.54 db
Filesize = 5821 Bytes (BP 7)



Benefits of the JPEG2000 Entropy Coder

- Bit plane representation of data allows for embedded bit stream
- Region of interest coding (ROI) coding can be performed by prioritizing the coding of the ROI bit plane information
- Arithmetic coding allows for efficient compression of sparse binary data
- Context modeling allows for efficient compression of binary correlated data
- Packetized information allows for improved error resilience

Example of Bit Plane Reordering



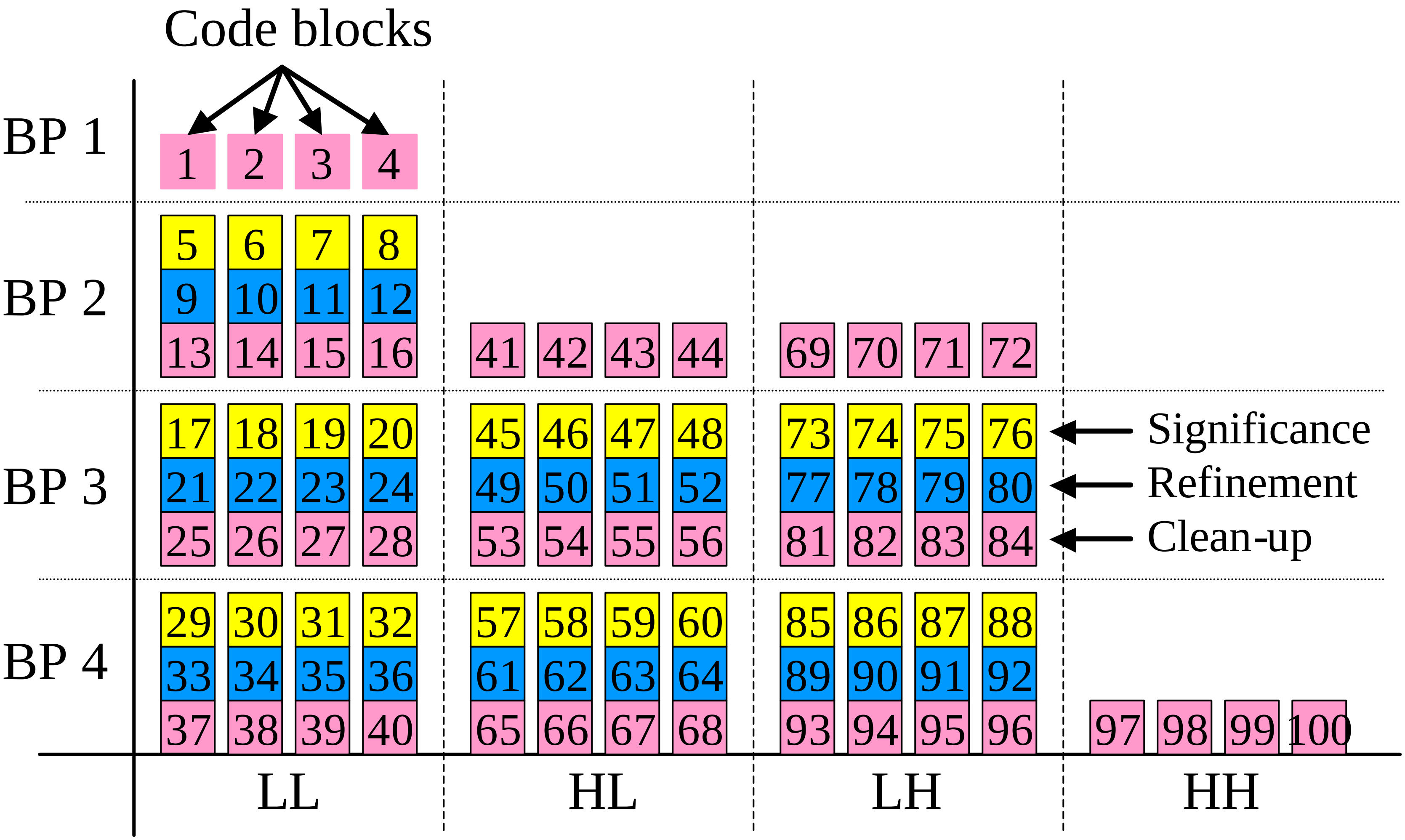
Subband 1

Subband 2

Subband 3

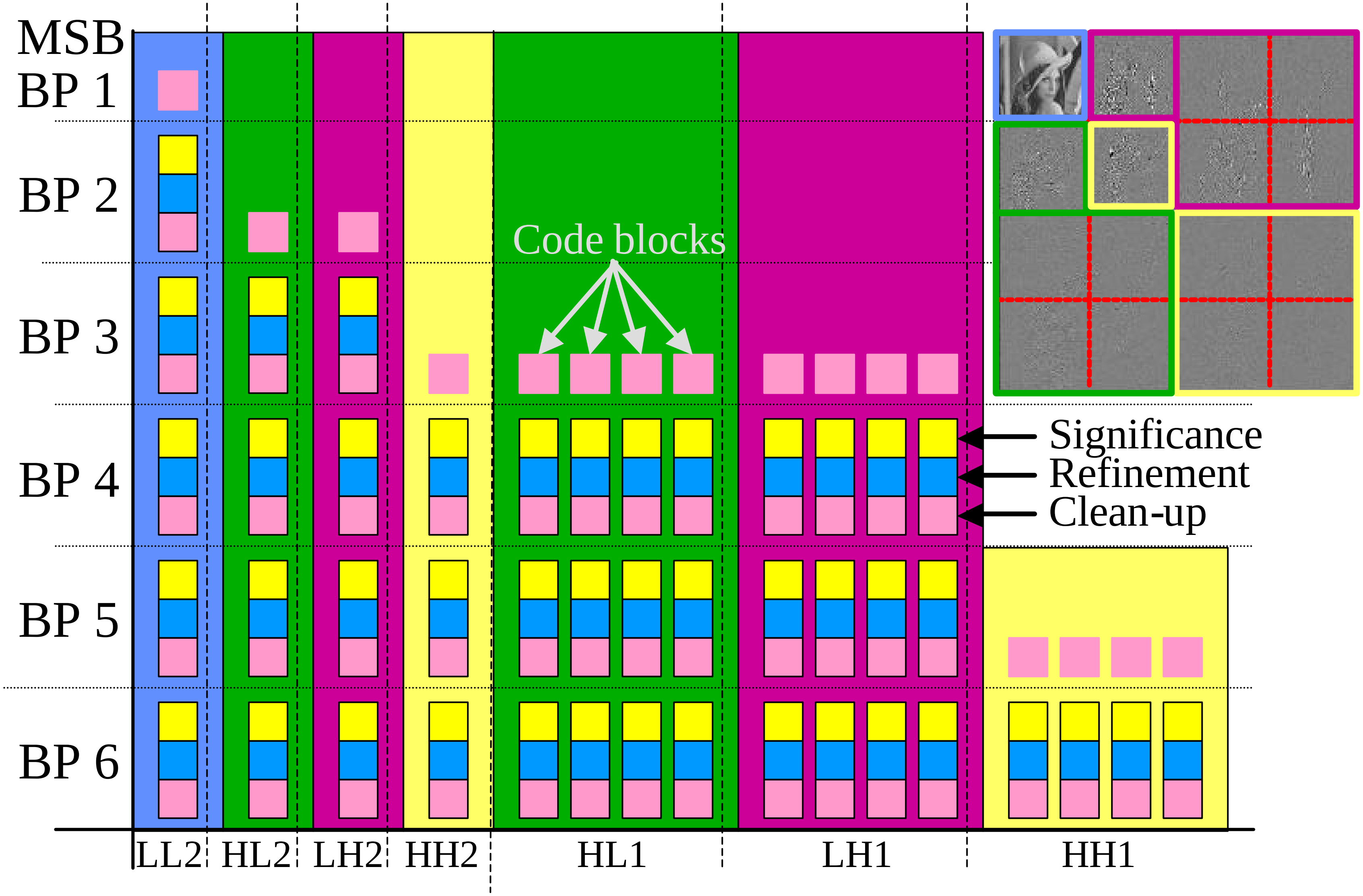
Subband 4

Example of Bit-Plane Reordering



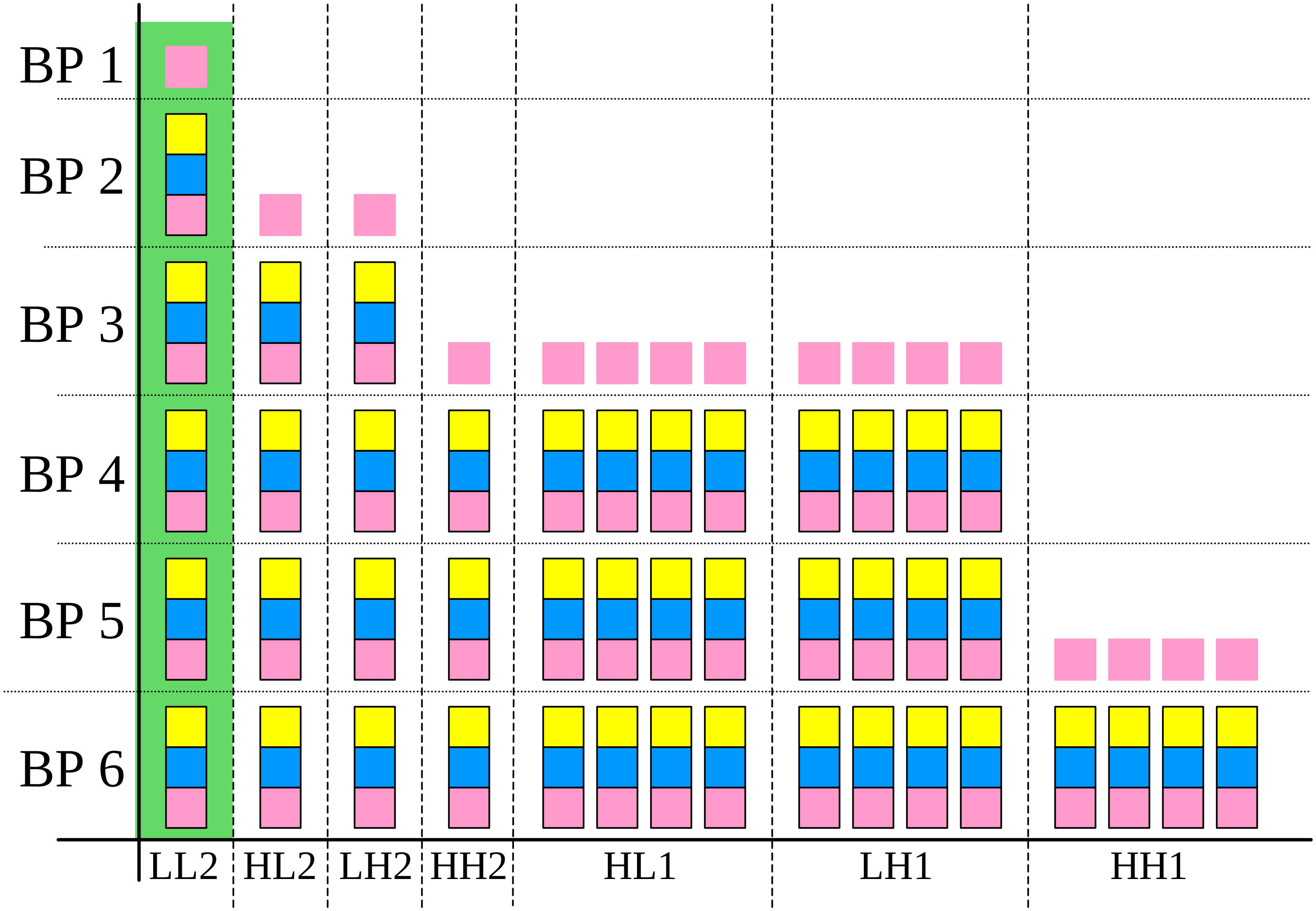
	LL	HL	LH	HH
BP 1	1 2 3 4			
BP 2	5 6 7 8 9 10 11 12 13 14 15 16	41 42 43 44	69 70 71 72	
BP 3	17 18 19 20 21 22 23 24 25 26 27 28	45 46 47 48 49 50 51 52 53 54 55 56	73 74 75 76 77 78 79 80 81 82 83 84	
BP 4	29 30 31 32 33 34 35 36 37 38 39 40	57 58 59 60 61 62 63 64 65 66 67 68	85 86 87 88 89 90 91 92 93 94 95 96	97 98 99 100

Example of Bit-Plane Data Ordering

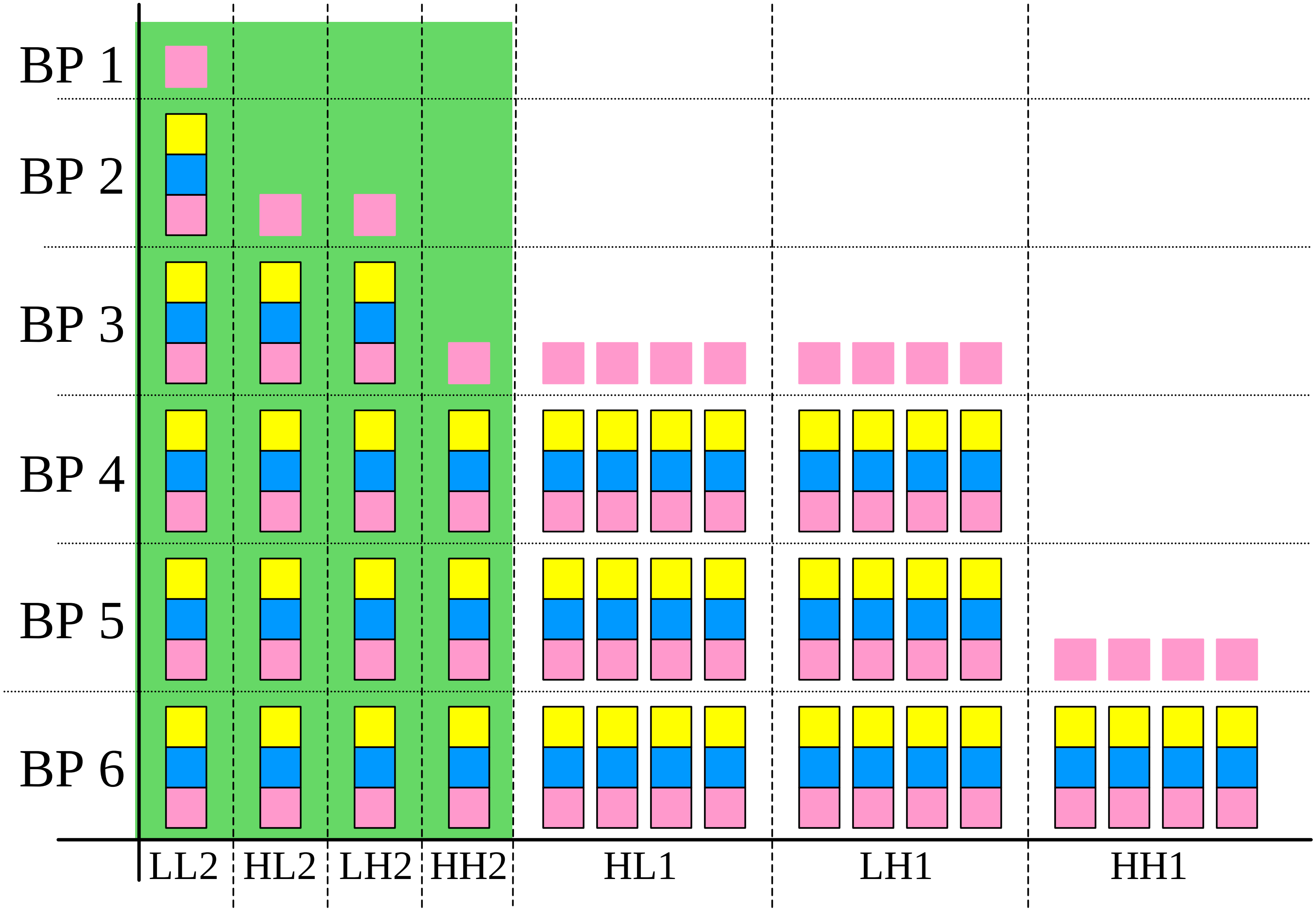


Systems Engineering Services

Lowest Resolution, Highest Quality

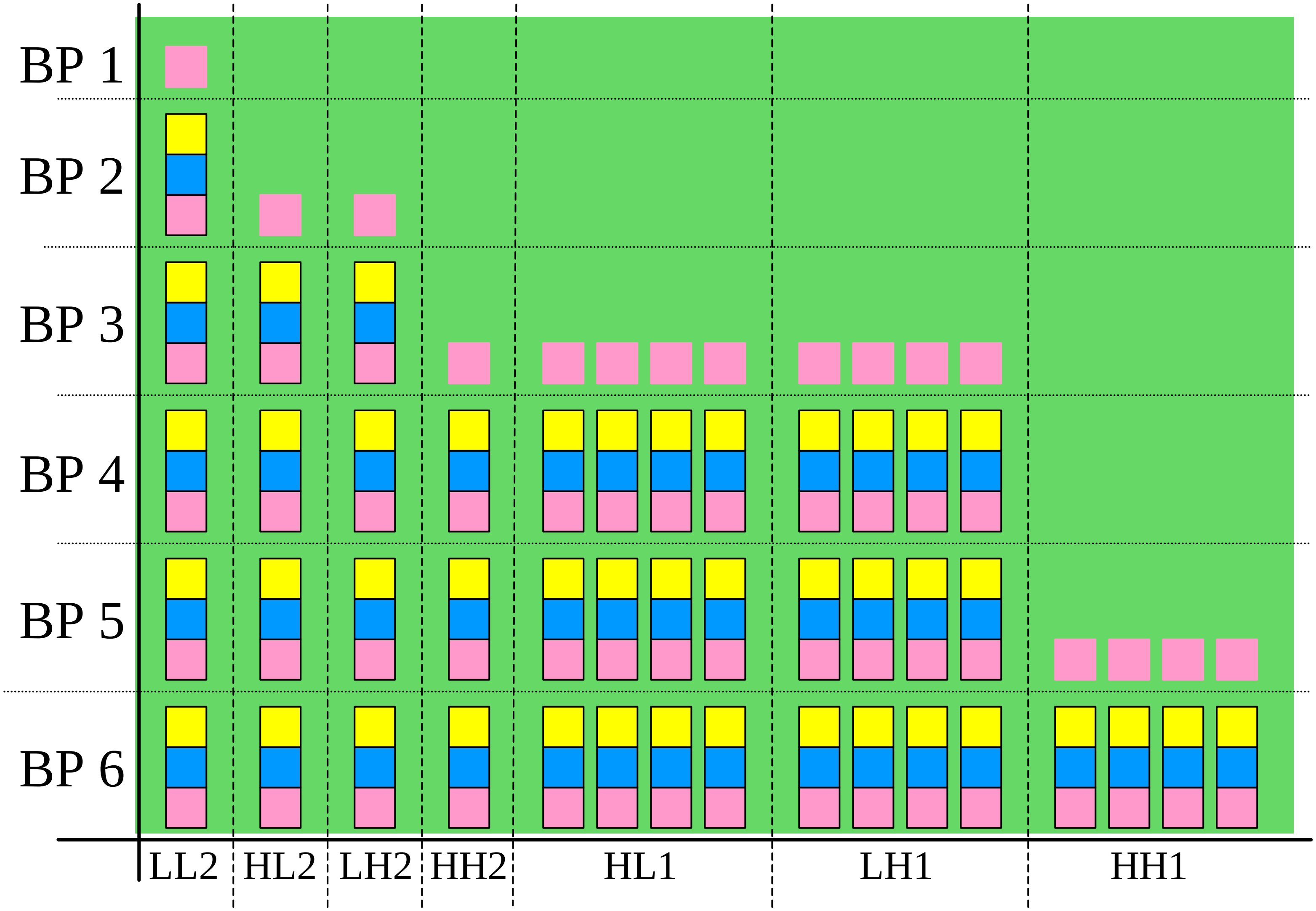


Medium Resolution, Highest Quality



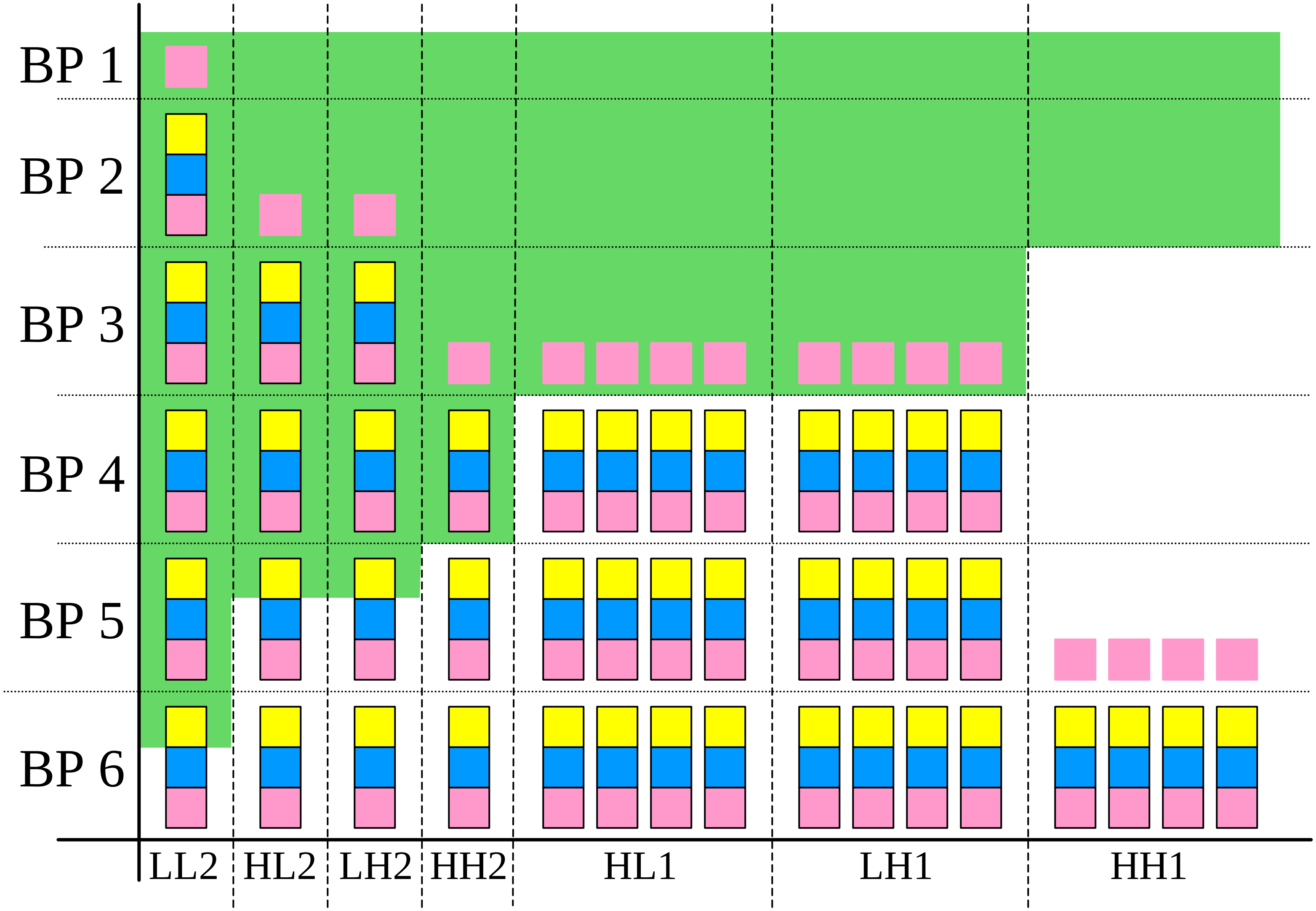
Systems Engineering Services

Highest Resolution, Highest Quality

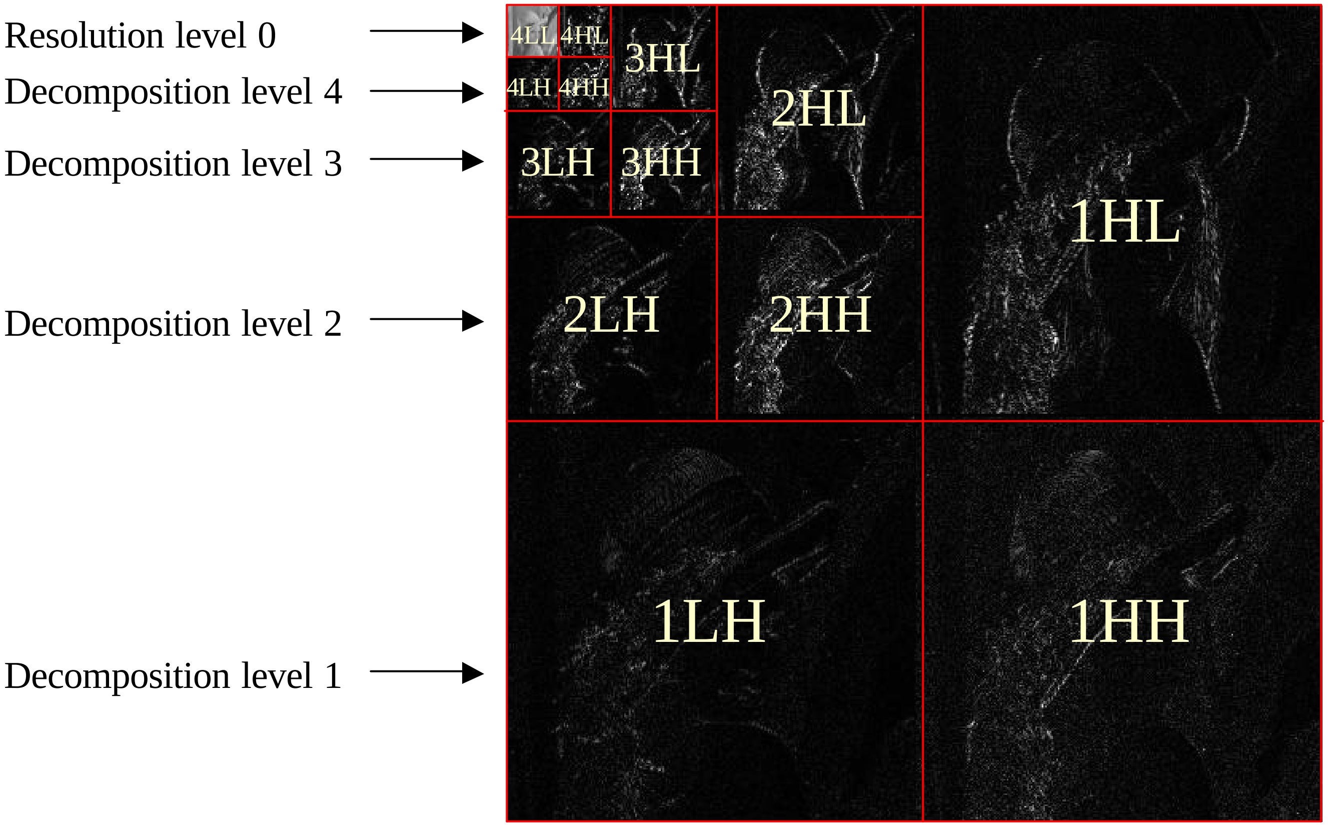


Systems Engineering Services

Highest Resolution, Target Visual Quality



Resolution & Decomposition Levels



Resolution level 0

Decomposition level 4

Decomposition level 3

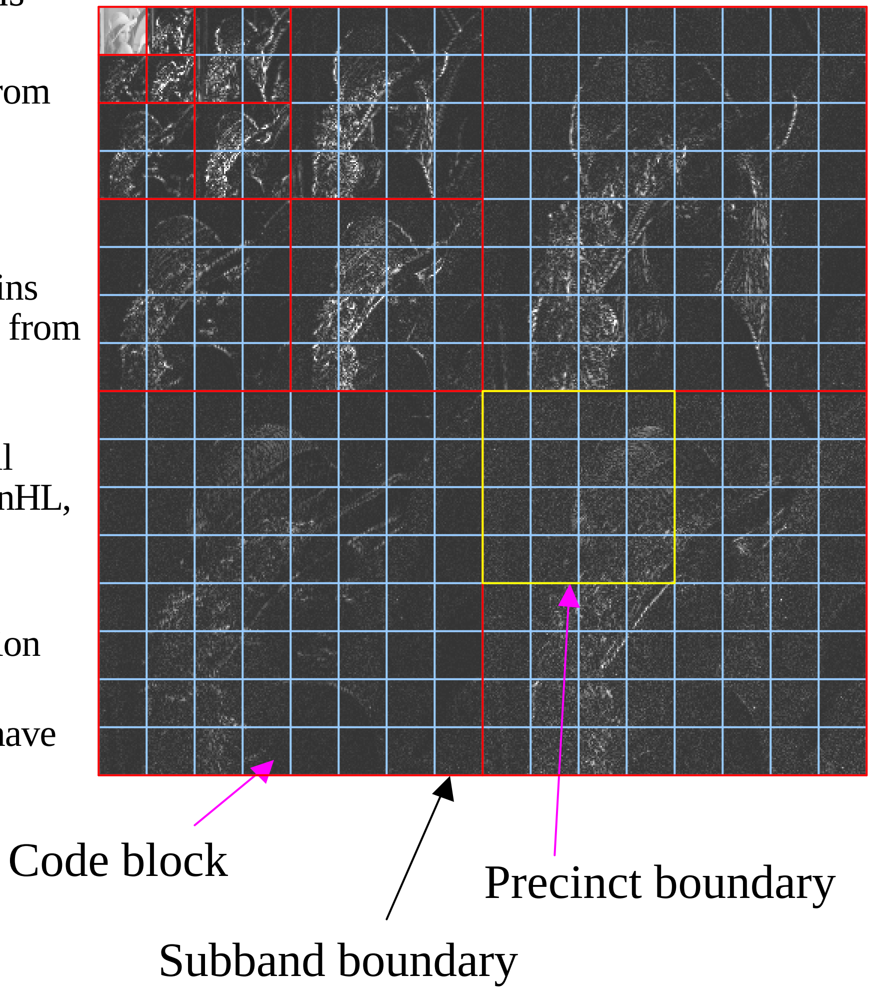
Decomposition level 2

Decomposition level 1

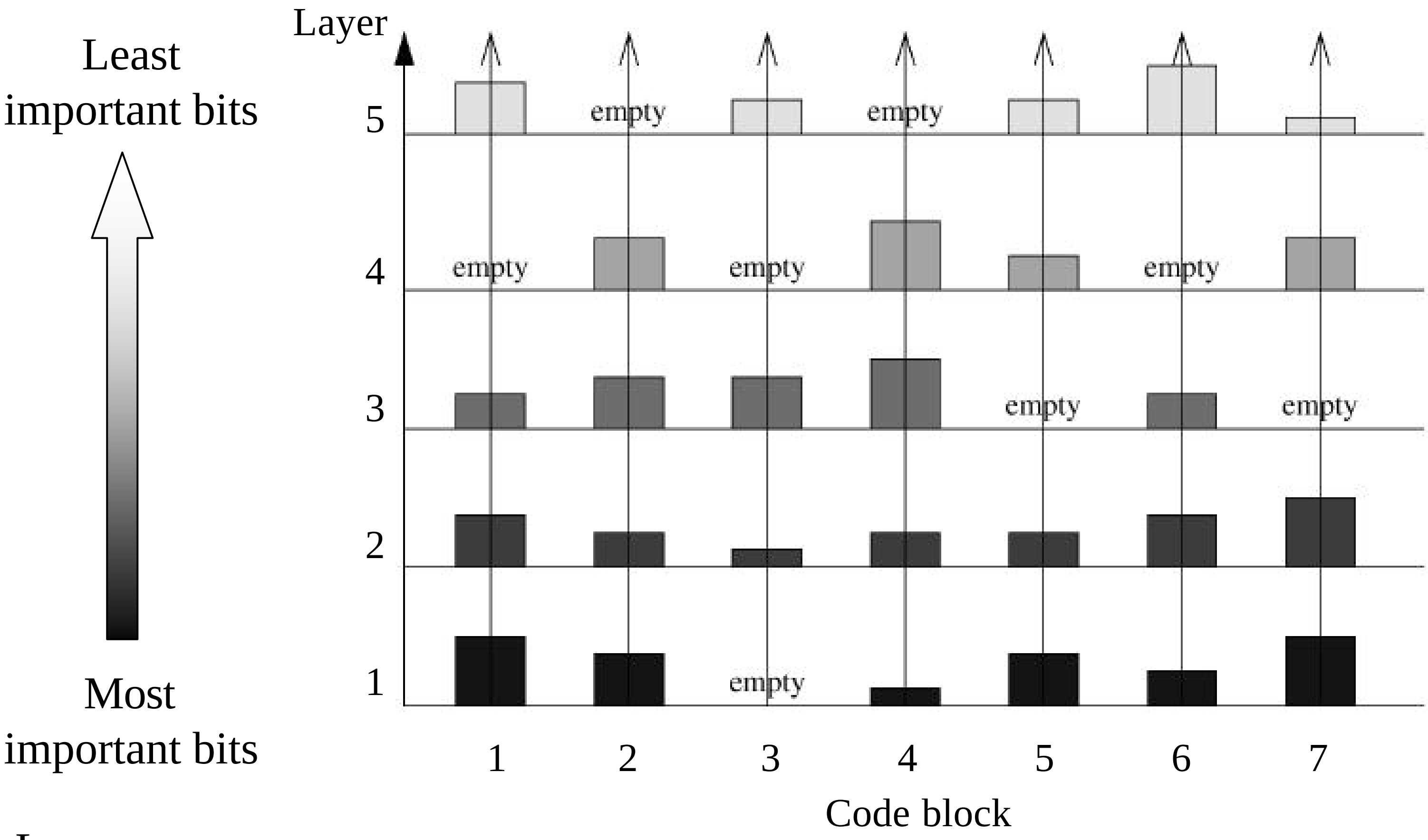
- Decomposition level
  – A set of HL, LH, and HH subbands. For the last level of decomposition (N_L), the LL subband is included. Decomposition levels run from 1 to N_L.
- Resolution level
  – Related to decomposition angle by $n = N_L - r$, where n is a decomposition level and r is a resolution level. Reconstruction levels run from N_L to 0.

Code Blocks, Packets & Precincts

- Code blocks divide subbands into regions that may be extracted independently
  - A rectangular grouping of coefficients from the same subband
  - Code blocks do not "scale" with levels
- Packets
  - Part of the compressed bitstream. Contains packet header & compressed image data from one layer of one precinct of one decomposition level
  - Contains compressed image data from all code blocks of a given resolution level (nHL, nLH, and nHH subbands)
- Precinct
  - Rectangular region within a decomposition level used to limit the size of packets
  - Subbands with more than one precinct, have multiple packets in a given layer.



Layer Formation

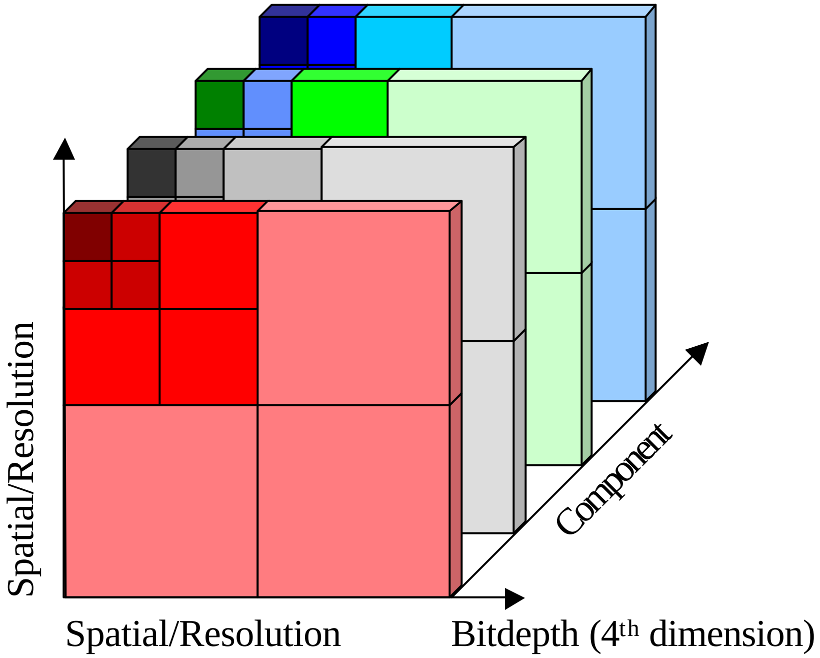


- Layer
  - An arbitrary collection of compressed image data from coding passes of one or more code blocks. Typically a layer represents an improvement in image quality.
  - Packet headers describe the contribution of each code block in each layer

Systems Engineering Services

JPEG 2000 Progression Types

- After wavelet processing, we have a four dimensional cube of data
  - Spatial/Resolution (two)
  - Component
  - Bitdepth
- JPEG 2000 allows progression along four dimensions
  - Layer (L)
  - Resolution (R)
  - Component (C)
  - Precinct or position (P)
- These are roughly equivalent as follows
  - Resolution & Precinct ⇔ Spatial/Resolution
  - Component ⇔ Component
  - Layer ⇔ Bitdepth



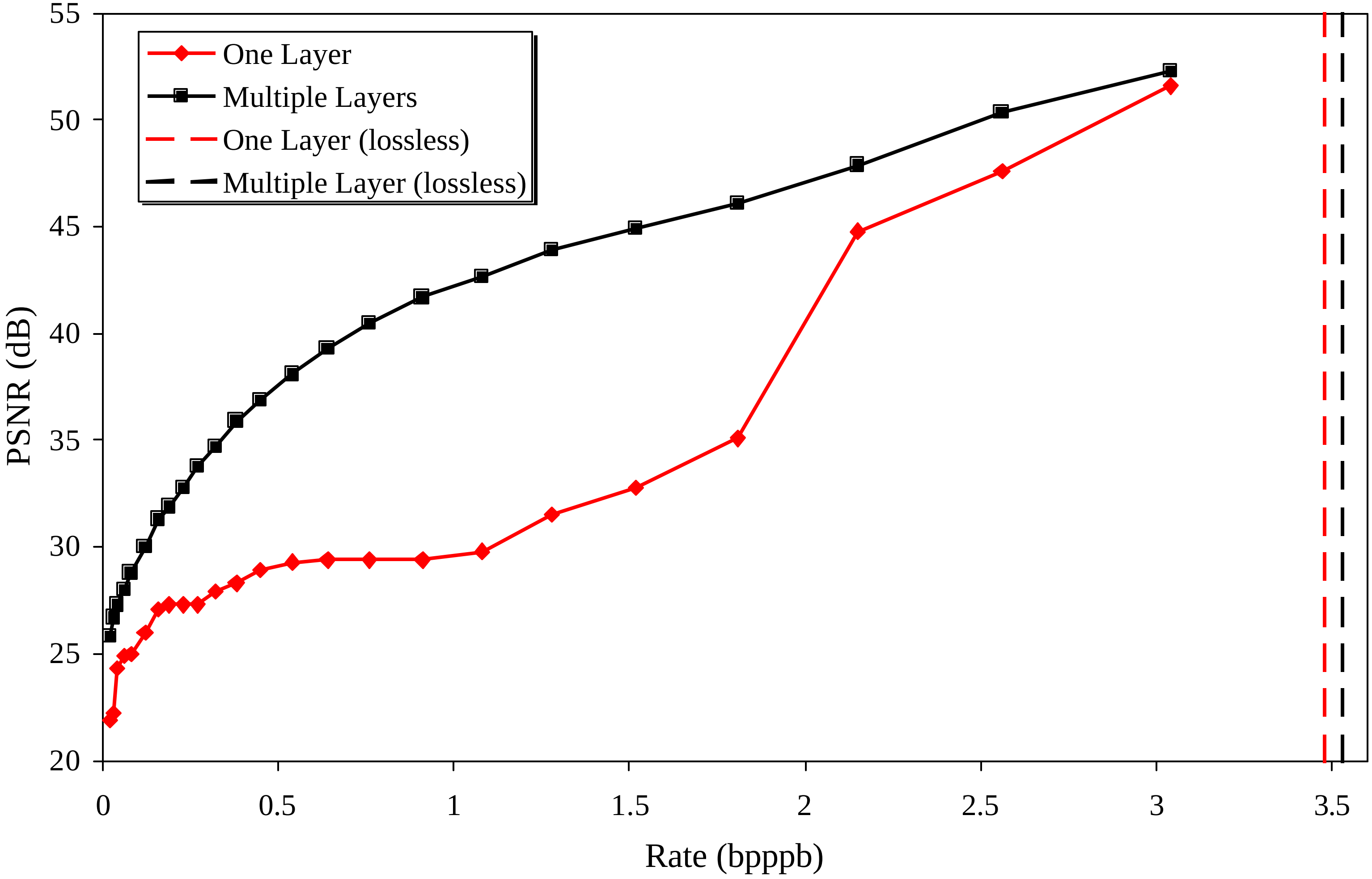
Wavelet processed components

JPEG 2000 Progression Types

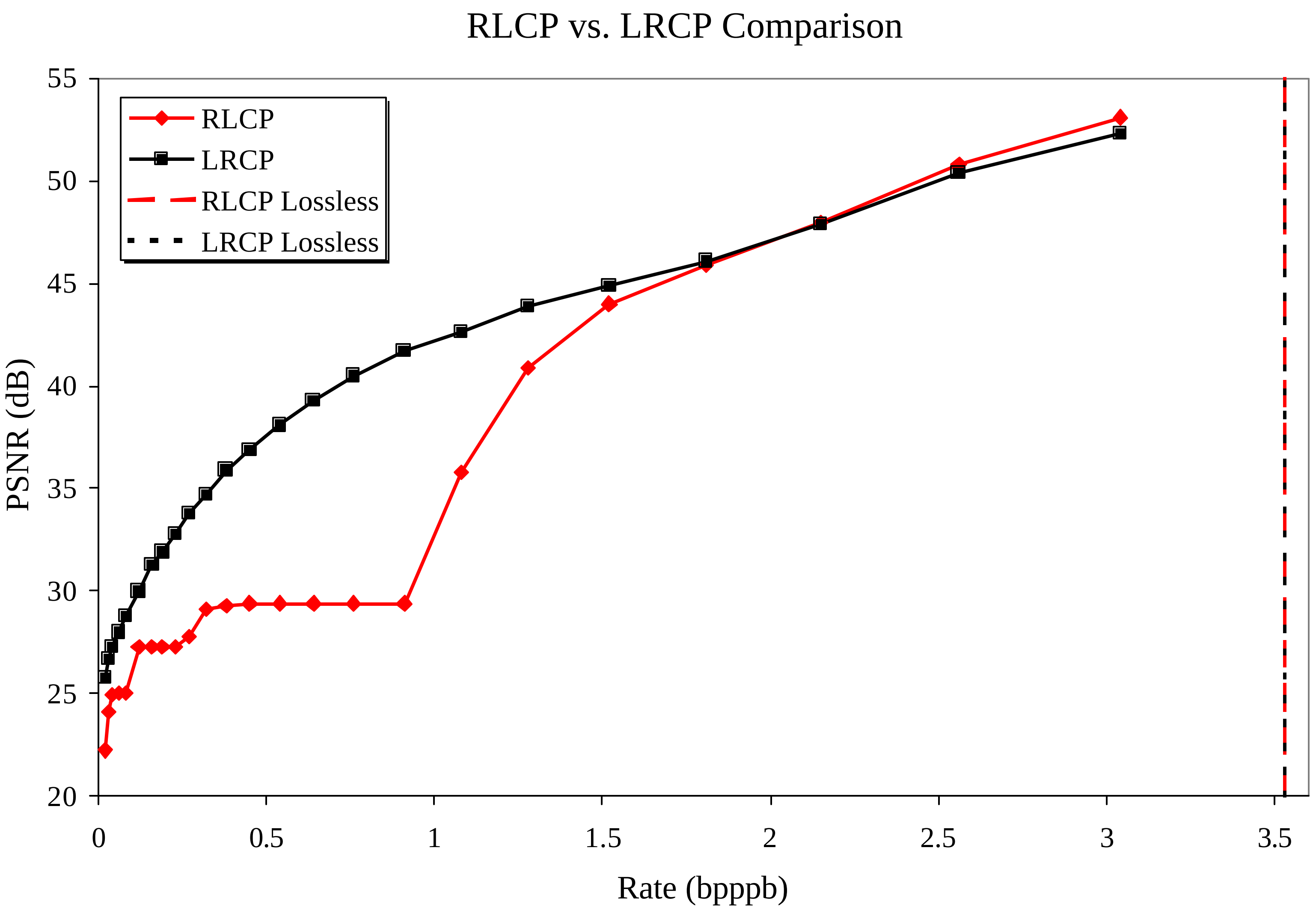
- There are five allowed progression types
  - LRCP, RLCP, RPCL, PCRL, CPRL
  - View progression as four nested loops, read left to right
- LRCP
  - Progression by SNR. Best full size image (SNR)
  - Loop over all layers
    - Loop over all resolution levels
      - Loop over all components
        - Loop over all precinct (positions)
- RLCP
  - Progression by resolution. Improving image quality for a fixed resolution
- Note that “C” (component) is in an inner loop in all progressions except CPRL

Effects of Layering

One Layer vs. Multiple Layers (LRCP)



Effects of Progression

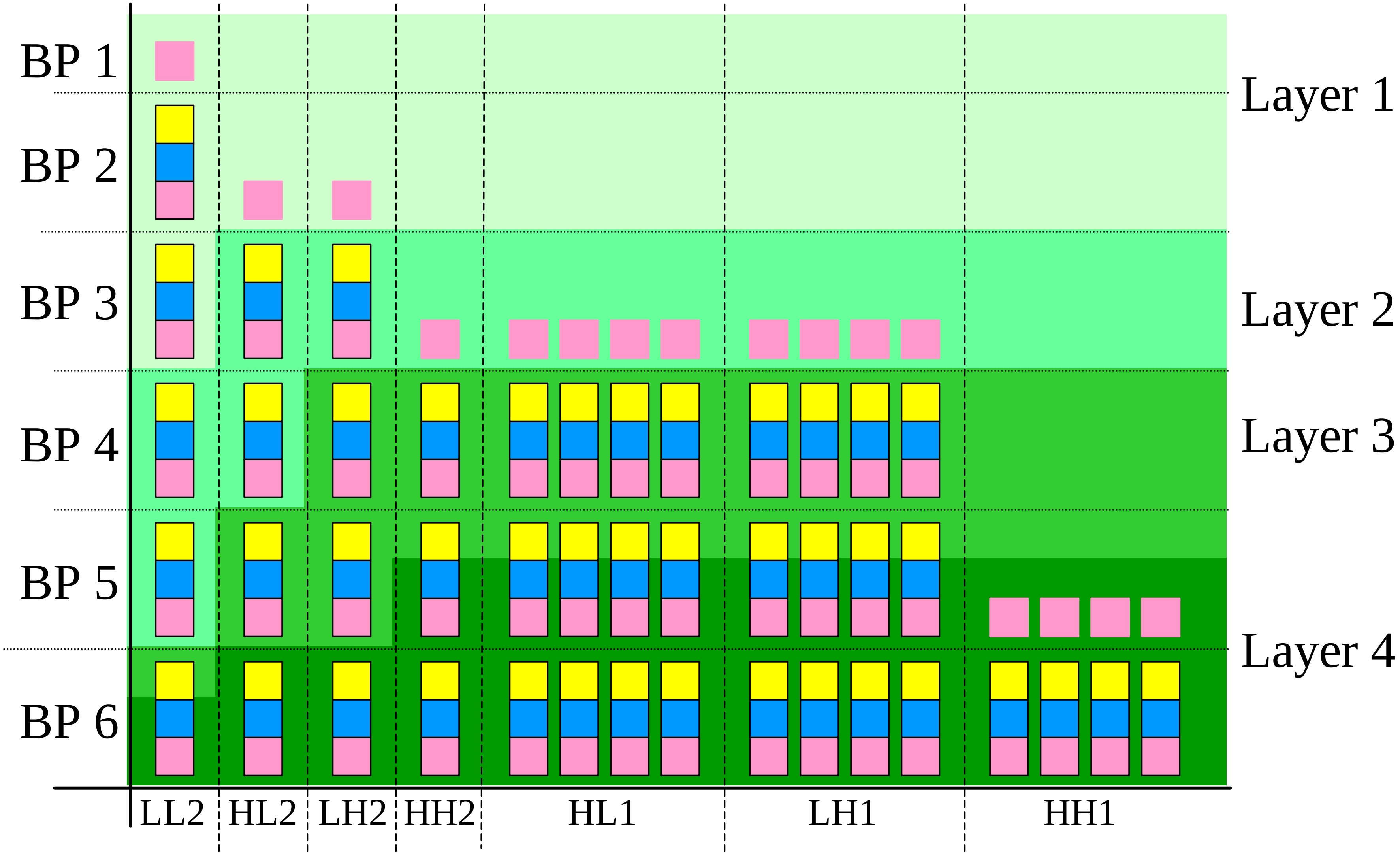


Tiles

- Tiles are independently coded sub images. Nothing crosses tile boundaries
  - Wavelet
  - Entropy coding
  - Layers
  - Progressions
- Tiles may be broken into tile parts. Tile parts from different tiles can be interspersed in a codestream
  - Only mechanism available to achieve “tile progression”
- In general, need to parse data out of tiles to achieve a different image quality
  - If all tiles are compressed at 2.0 bpp and you want 1.0 bpp, then need to go into each tile and get the 1.0 bpp

NIMA
Systems Engineering Services

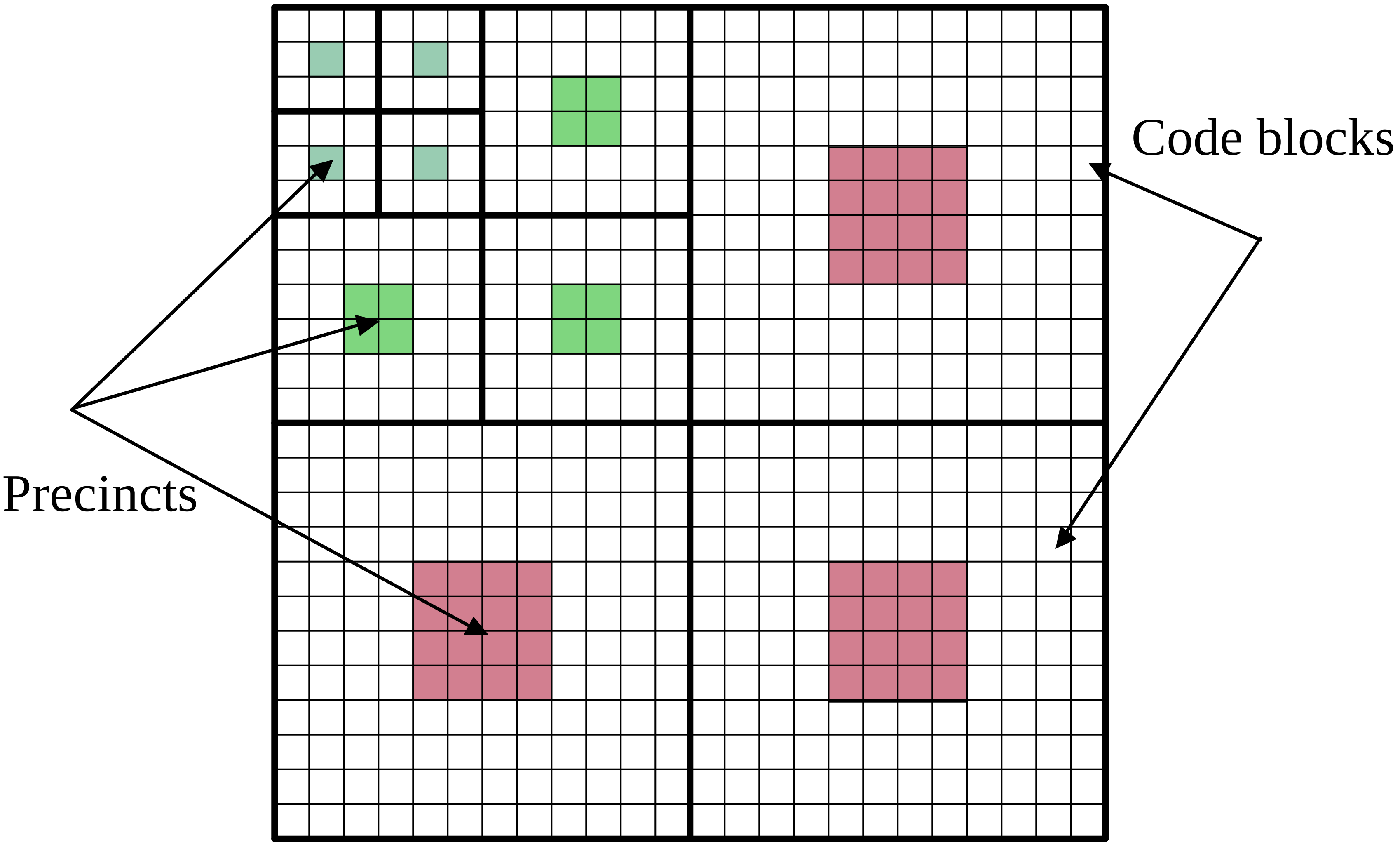
Bitstream Ordering with Four Quality Layers



JPEG2000 Bitstream Syntax

- **Precinct**: Each subband is divided into non-overlapping rectangles of equal size (except maybe at the boundaries where the size can be different) called precincts. Precincts provide some level of spatial locality in the bit stream and their boundaries are aligned with code blocks. Their size can vary for each tile, (color) component, and resolution.
- **Packet**: Consists of the compressed bit streams associated with a certain number of sub-bit planes from each codeblock in a precinct. Packets serve as one quality increment for one resolution level at one spatial location.
- **Layer**: is a collection of packets, one from each precinct at each resolution. It can be interpreted as one quality increment for the entire image at full resolution.

Example: Precincts and Codeblocks

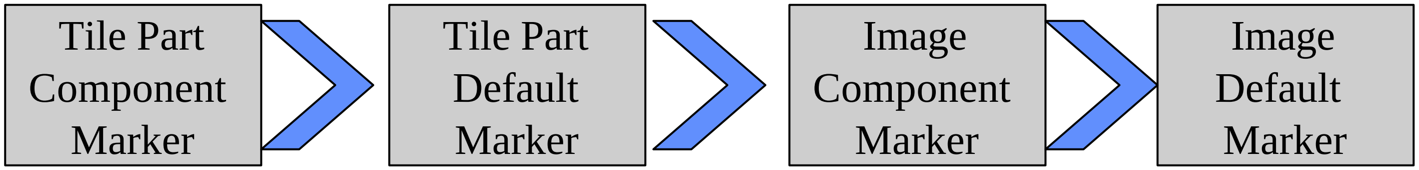


JPEG 2000 Codestream Structure

- Delimiting markers
  - These are markers that are used in the start of most major sections of codestream and the very end.
  - Start of codestream (SOC), start of tile part (SOT), start of data (SOD), and end of codestream (EOC)
- Fixed information marker segments
  - This marker includes information that is required to properly decode the image.
  - Size Marker (SIZ) which includes reference grid size, tile size, resolution/sampling (relative to grid), image and tile offsets (into the grid), number of components, and component precision (data type and bit depth)

JPEG 2000 Codestream Structure

- Functional Marker Segments
  - These marker define the parameters used in the compression of a tile or an image. The order or precedence of the markers are



Tile Part Component Marker → Tile Part Default Marker → Image Component Marker → Image Default Marker

  - Default markers (can be used in image header or tile header)
    - coding style default (COD), quantization default (QCD), region of interest (RGN), progression order changes (POC)
  - Component Markers
    - coding style component(COC), quantization component (QCC)

JPEG 2000 Codestream Structure

- Functional Markers
  - Coding style default and coding style component include information on coding style, number of decompositions, code-block size, code-block style, wavelet transformation, precinct style
  - Quantization default and quantization component include information on the quantization for the derived, expounded or no quantization
  - Region of interest marker includes the location of the ROI
  - Progression order changes describes the bounds and progression order for any progression other than that in the COD marker in the main image header

JPEG 2000 Codestream Structure

- Pointer Marker Segments
  - Pointer markers are used for quick access to data that is required for decompression of a given location, resolution quality, or component.
  - All of the pointer segments define lengths of segments which allow fast rearranging of data and pointer markers
  - Main Header Markers
    - Tile part lengths (TLM), packet lengths main header (PLM), packed packet headers (PPM) (main header).
  - Tile part header
    - Packet length tile-part header (PLT), packed packet headers (tile part header) (PPT).

JPEG 2000 Codestream Structure

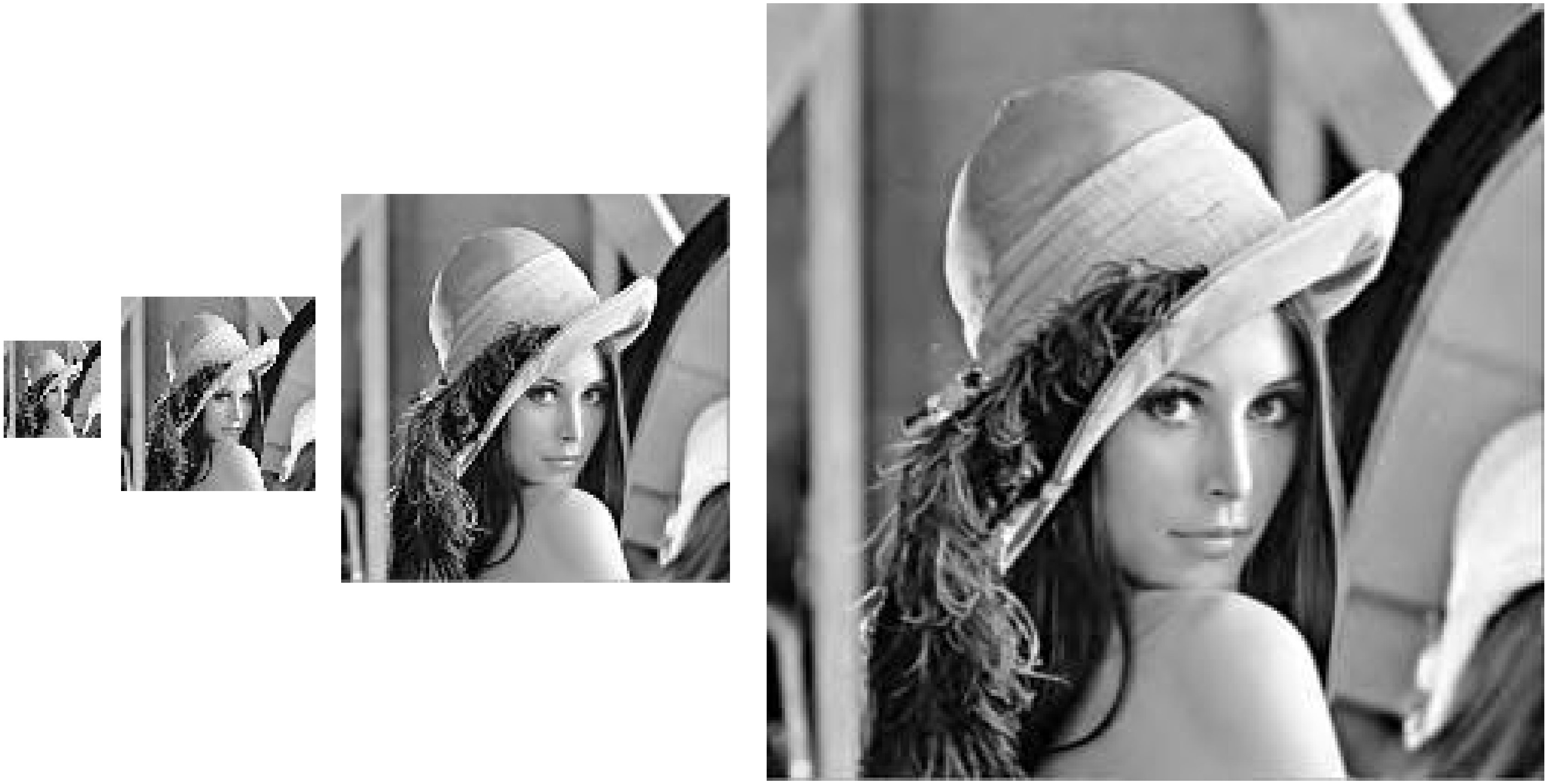
- In Bit stream markers
  - Start of packet (SOP) and end of packet (EOP) markers are used to isolate a given packet in a noisy environment
- Information marker segments
  - Component registration (CRG) marker is used to register components if the components do not have the same sampling to the reference grid
  - The comment marker (COM) is an open style marker that allows for unstructured data

JPEG 2000 Codestream Structure

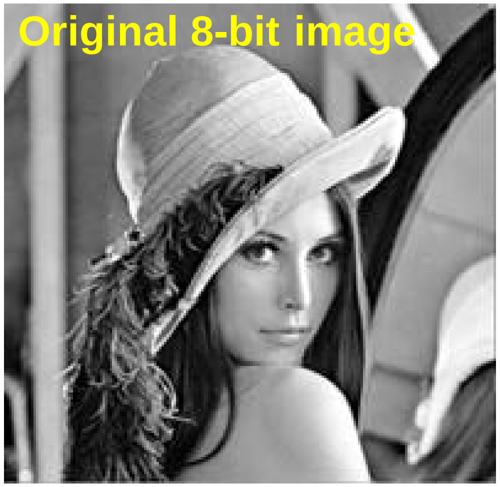
- The J2K codestream starts with the main header, followed by tile-part header(s), and bitstream(s) and ends with an EOC marker
  - The main header starts with SOC and SIZ markers, then followed (in any order) by COD and QCD markers and possibly QCC, RGN, POC, PPM, TLM, PLM, CRG, COM
  - The tile part header starts with SOT marker and finishes with SOD and can contain (in any order) COD, COC, QCD, QCC, RGN, POC, PPT, PLT, COM
  - The bitstream may contain SOP and EPH markers

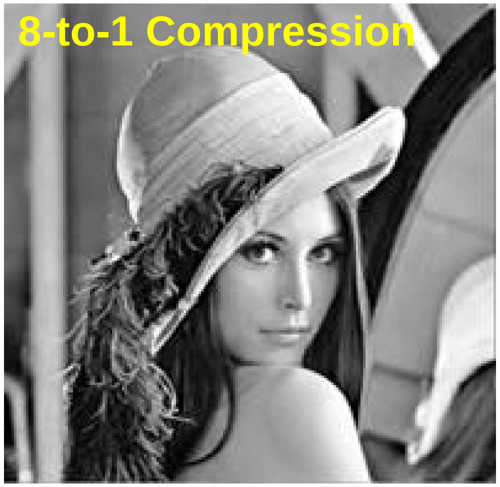
Resolution Progressive Example

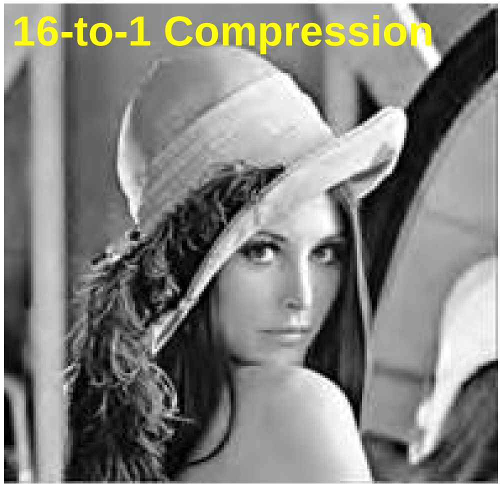
- All images have been decompressed from the same bit stream. The wavelet decomposition provides a natural resolution hierarchy.

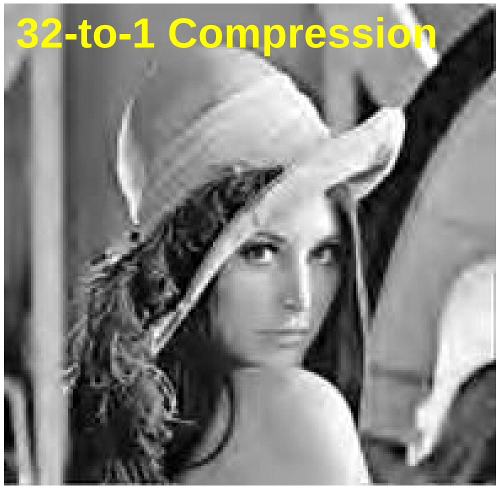


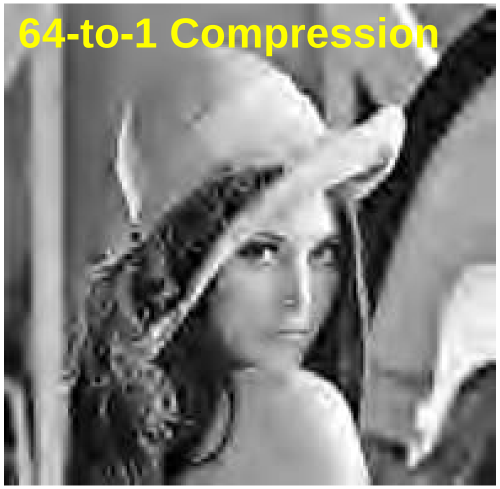
SNR Scalability Example

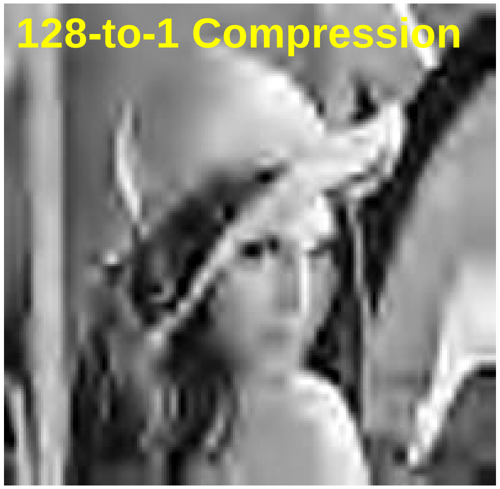

Original 8-bit image


8-to-1 Compression


16-to-1 Compression


32-to-1 Compression


64-to-1 Compression


128-to-1 Compression

All images have been decompressed from the same bit-stream

Spatial Progression Example

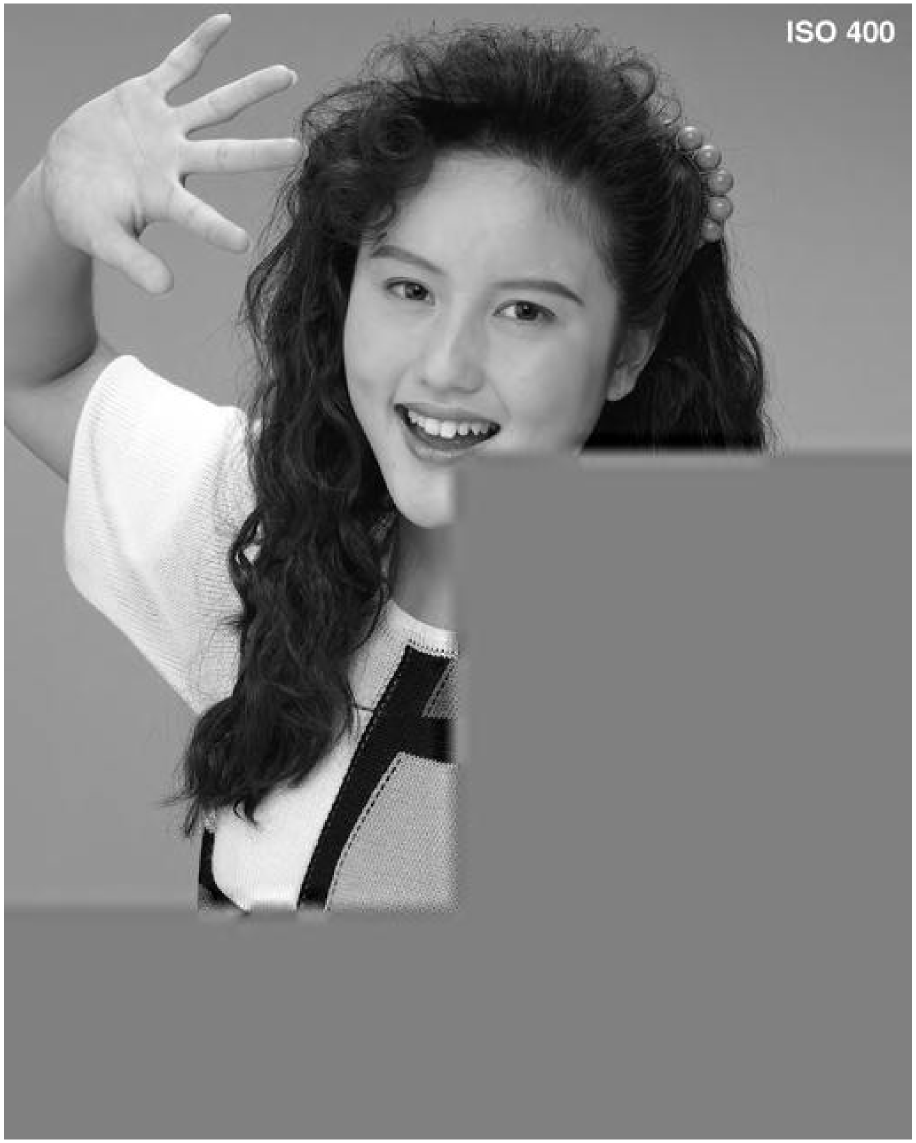


Image after decoding 3 precincts using the component-precinct-resolution-layer ordering.

Precinct sizes are chosen such that each subband precinct will correspond to a 1024x1024 image region. The image is (1024x2560), so there are 2 partial precincts at the bottom of the image.

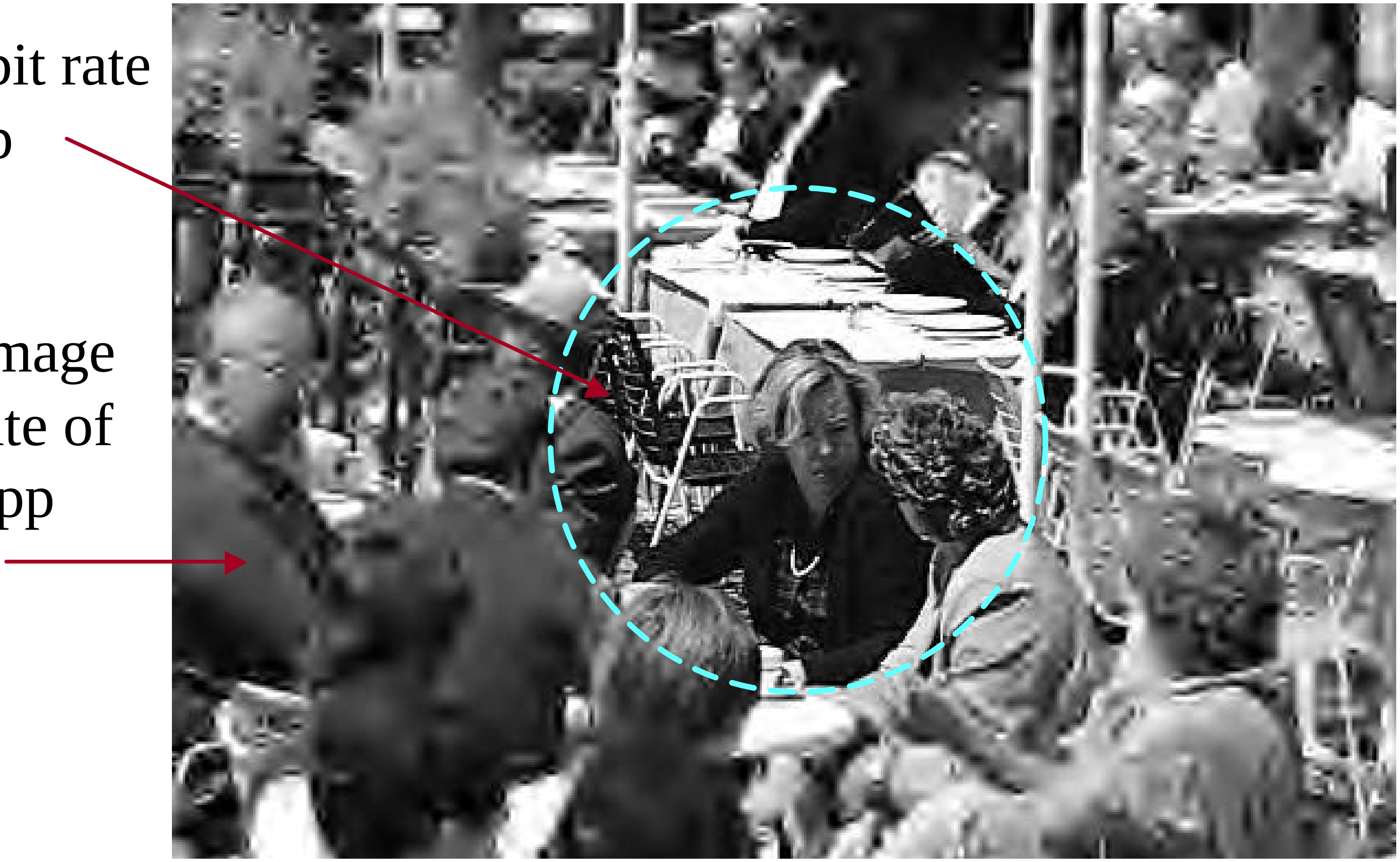
Region of Interest (ROI) Coding

- **Region-of-interest** (ROI) coding allows selected parts of an image to be coded with higher quality compared to the background (BG).
- The ROI encoding is done using an **ROI mask**. This binary mask is generated in the wavelet domain and describes which quantized wavelet coefficients must be encoded with higher quality. It depends on:
  - ROI region specification in the image domain
  - DWT filter

ROI Coding in Part II

- Part II allows for the scaling method of ROI coding, where the bits representing the wavelet coefficients contributing to the ROI are shifted upward by a user-defined value. This
  - Allows for coding the ROI with any desired quality differential compared to the background.
  - Multiple ROI's are allowed, each with its own corresponding scaling value.
  - The ROI shape is limited to rectangles and circles. ROI coordinates and shift values are signaled in the bitstream.
  - The ROI mask needs to be generated both on the encoder and the decoder side.

Region of Interest (ROI) Example



ROI has bit rate of 2.0 bpp

Rest of image has bit rate of 0.0625 bpp

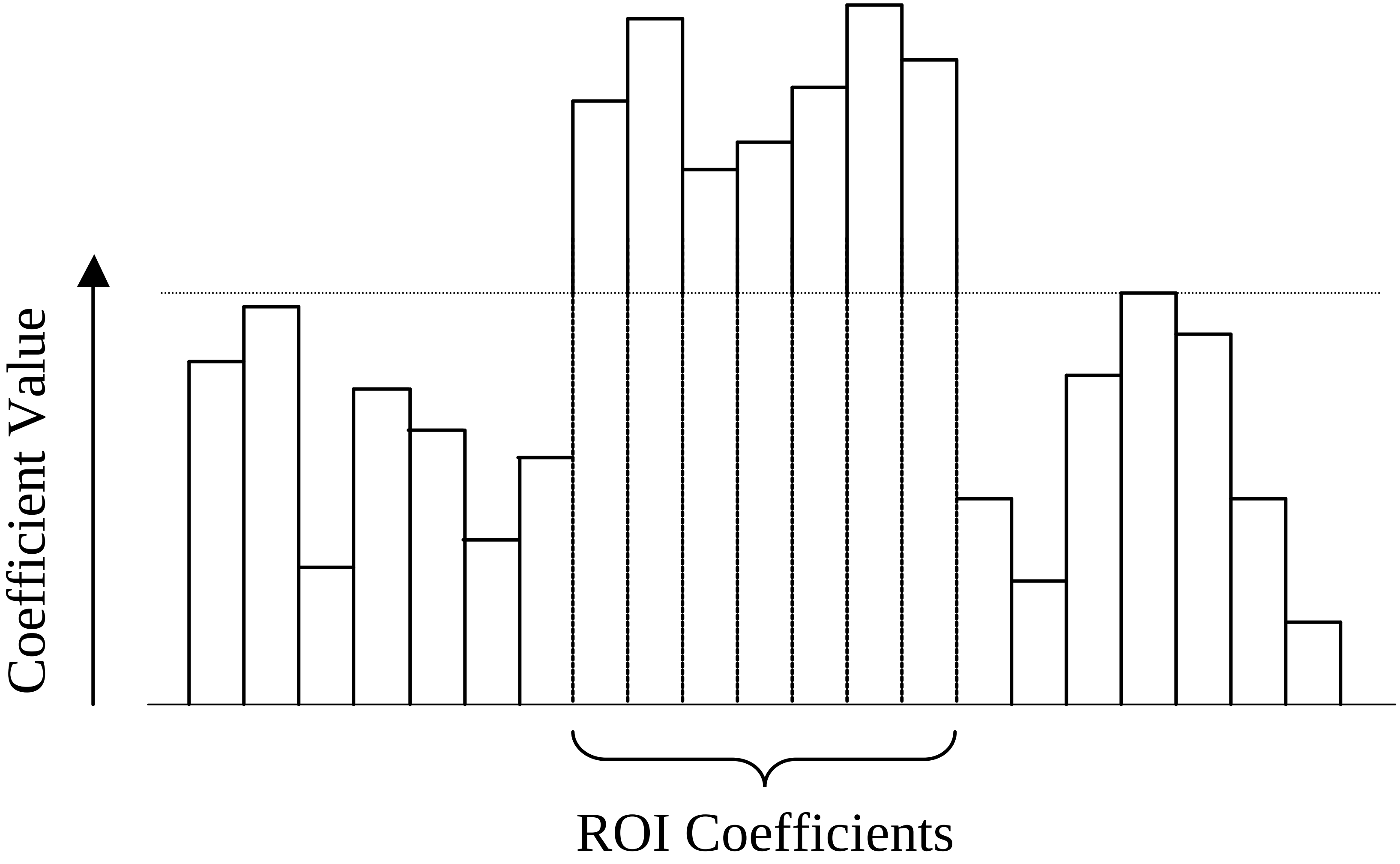
Bit rate for entire image is 0.12 bpp

ROI Coding in Part I: Maxshift Method

- In the **Maxshift** method, the wavelet coefficients in the ROI region are scaled up by a fixed number of bits *s*, so that the smallest shifted nonzero ROI coefficient is larger than the largest BG coefficient. The parameter *s* is signaled in the bit stream. As a result, the decoder can discriminate between the ROI and BG coefficients by comparing each decoded value to a threshold. Pros and cons are:
  - It allows for multiple regions of arbitrary shape ROI without the inclusion of the shape information in the bit stream or need for ROI mask generation at the decoder.
  - The user can prioritize the coding of the ROI region over the BG but does not have control over the quality differential between ROI and BG.

Systems Engineering Services

Maxshift Method of ROI Coding



Other JPEG2000 Features

- Tiling of large images provides for independent processing of different image regions.
- The canvas coordinate system allows for efficient recompression of cropped images.
- Rich bit stream syntax provides means for transcoding of the data for streaming, resolution progression, quality progression, or any mixture thereof.
- Rich file format allows for various color spaces and metadata information.
- Compressed domain image manipulation (rotations of 90, 180, 270 degrees, horizontal and vertical flipping)

JPEG2000 Feature Summary

- Improved coding efficiency (up to 30% compared to DCT)
- Multi-resolution representation
- Quality scalability (SNR or visual)
- Target bit rate (constant bit rate applications)
- Lossless to lossy progression
- Improved error resilience
- Tiling
- Rich bit stream syntax (layering, packet partitions, canvas coordinate system, etc.)
- Rich file format